A DATA-FIRST PRESERVATION STRATEGY: DATA MANAGEMENT IN SPAR

Louise Fauduet
Bibliothèque nationale de France
Department of Preservation and Conservation

Sébastien Peyrard
Bibliothèque nationale de France
Department of Bibliographic and Digital Information

ABSTRACT

The Bibliothèque nationale de France has developed its trusted digital repository, SPAR (Scalable Preservation and Archiving Repository), as a data-first system. This implies having fully described collections, through use of metadata standards in the information packages, such as METS, PREMIS, MIX or textMD, in a way that will make sense given the diversity of our documents.

The need for full documentation also applies to the system itself. On the one hand, SPAR is self-describing in order to ensure its durability. On the other hand, all the information that is ingested into the system contributes to determine its settings and its behavior. The Data Management module is at the heart of these information flows.

We expect to push this data-first objective ahead by using RDF technology, based on existing and trusted information models and ontologies, such as OAIS and PREMIS. The challenges and successes we encounter all serve the greater goal of having a unique and versatile data model for every user of the system, whether collection curator or system manager.

1. INTRODUCTION

The SPAR system, Bibliothèque nationale de France's trusted digital repository, is finally stepping out of the design phases and becoming a concrete tool in preservation and collection management at the BnF (See Bermès and al. [1]).

SPAR is conceived as a data-first system, where data is used both to curate the collections and to manage the system.

The collections are fully described and each piece of information should be individually accessible. The flexibility in querying information is intended to make collection management as easy as possible from a preservation perspective.

The system is fully self-describing: every process is documented within it; and it can be set up by the ingested data, without having to change the actual implementation of the system.

The data-first approach is a three-part endeavor. First, it depends on the way the OAIS information model is implemented in the information packages. Then, it relies on the translation of the OAIS functional model into SPAR's architecture, making the Data Management module possible. Last but not least, the use of RDF enables BnF staff to draw on the data in order to manage the system and the collections.

2. DESCRIBING THE COLLECTIONS: OUR METADATA STANDARDS

2.1. The Metadata Makings of an AIP

2.1.1. METS: the Why

Each digital document is ingested into the SPAR preservation system as an Information package, as defined by the OAIS model, with a METS manifest as packaging information stored within each package. Expressing our information needs in a standardized way and in compliance with best practices facilitates maintenance and is therefore a great ally in digital preservation.

METS, like other preservation metadata formats, offers great flexibility, and many further choices are required in order to implement it — which sections to use, which other metadata formats to embed, which granularity levels to define in order to describe the package, and so on.

The challenge of these numerous implementation choices prompted librarians to reflect on best practices which would fit the BnF's specific needs without reducing interoperability[1], even if actual exchange between repositories is not in our short- or medium-term plans. One of the greatest advantages of METS is indeed its wide use in the digital preservation world in general and in libraries in particular. Its active user community facilitates METS's implementation while protecting against format obsolescence.

2.1.2. METS: the How

The abstract quality and great genericity of OAIS along with the flexibility and openness of METS made the implementation of both in the BnF context a great step in itself. The main choices that had an impact on the coverage of the collections by the metadata, involve METS sections, granularity levels, and embedded information.

[1] On this issue, see for instance Rebecca S. Guenther [4].

First, we chose to exclude from our METS implementation the metsHdr, structLink and behaviorSec, for which we had no need, and the rightsMD subsection, since we would rather have a dynamic calculation of the legal status of a document at the time it is accessed (See Martin [5]).

The main factor in the choice of granularity levels in METS's structural map was the great diversity of material to be ingested in SPAR: digitized texts and still images at first, then digitized and born-digital audiovisual content, Web archives, the library's born-digital archives, and so on. The adoption of generic terms to describe the levels within the digital object avoids the heavy maintenance of a specific vocabulary.

Therefore, four levels were adopted in the structural map. From the broader to the narrower, they are:

- set: ensemble of groups. This level is only intended to contextualize groups by describing a higher level, which is purely intellectual. E.g. serial, or multivolume monograph.
- group: the reference level in our repository. It is the level at which a digital document is digitized and/or manipulated. E.g. physical volume of a monograph; CD…
- object: an intellectual subdivision of a package E.g. page of a document, side of a vinyl…
- file: a concrete file.

Regarding embedded schemes in the dmdSec and amdSec sections of METS, three main decisions were made.

Dublin Core is implemented in dmdSec and sourceMD: using METS from a preservation perspective, we don't need to include in AIPs the type of highly structured descriptive information that already exists in our catalog[1]. This type of information shows what the package is about, but is independent of the actual digital embodiment of the document; it is not needed to make preservation plans. More pragmatically, its non inclusion in the packages avoids close dependencies and mutual updates between two systems, our catalog and SPAR, so that the Archive is as autonomous as possible.

However, some specific information needs, expressed by SPAR's users at the librarian end, require more elements than the DC's 15 standard ones: description of the institution detaining the files requires Qualified DC; domain specific identifiers such as ISSN, ISBN, bar code, call numbers or even pagination types required more specific elements that did not exist as such in DC; so we used our own schema, adding as few elements as we could. This infringement on our interoperability vow is a compromise that enables a better management of librarian needs.

[1] http://catalogue.bnf.fr/

Finally, we use premis:object and premis:event in the techMD and digiprovMD sections of METS, because of PREMIS' genericity and closeness to the OAIS, and of the wide adoption of the "METS + PREMIS" duo among libraries.

However, premis:object is not intended to express text-, image-, sound- and video-specific file characteristics. To this end, we use the METS-proof and widely adopted MIX scheme for image files and textMD for text files.

An overall consensus on a characterization format for audio, and above all video content, has yet to be reached in the digital preservation community. Few schemas are able to express every piece of information our audiovisual experts need for collections management in a well-structured and thus easily manageable form. Conversely, few are designed to be used inside packaging information, and thus make elements we express in other sections of METS mandatory.

Our double need of expressivity and modularity brought us to MPEG-7, an ISO standard, suited to both audio and video, and even to multimedia and program files. Therefore we rejected more widely adopted standards for audio files, such as AES-X098B.

2.2. Describing a preservation system with data: reference information packages

The choices we made regarding METS define our SIPs, AIPs and DIPs in a way that satisfies our information needs as to the digital documents we preserve. Yet there is an equally important type of information that also has to be preserved in SPAR: all the documentation regarding the way the system works and the nature of the information that is preserved in it. In order for SPAR to be self-referenced and OAIS-compliant, this information is enclosed in information packages as well, in a category that we named reference information packages. They can be of three different types: context, formats and agents.

Context reference information allows us to create links between ensembles of packages that share certain characteristics. In SPAR, this mainly means assigning packages to their relevant track and channel. A track is a family of documents with similar intellectual and legal characteristics: there is digitized printed content track, a Web legal deposit track, and so on. Each track is divided into channels, which share homogeneous technical characteristics[2]. Description of each channel and track is factorized in a dedicated information package. In the future, we intend to use information packages to describe software environment in an emulation perspective.

[2] For instance, the channel B of the Audiovisual track contains the product of the digitization of analog audio and video document acquired through legal deposit, with well-described and easily manageable production formats; whereas the channel A of the same track concerns legal deposit of born digital content (excluding documents harvested on the web), which we are constrained to ingest "as is", with inevitably unknown or misused formats.

We also give representation information about every format for which we have designed a preservation strategy. This can include standards such as TIFF 6.0, or BnF profiles restraining these formats, for instance uncompressed 24 bits TIFF in 300 dpi resolution.

Finally, SPAR ingests reference information about agents performing preservation operations, which can be human (administrator, preservation expert), software tools (identification, characterization and validation tools) and processes in SPAR (such as the ingest and package update process).

Grouping information that is common to many digital objects is just one feature of reference packages. They have maintenance enhancement advantages: updating this central information means it is not necessary to update every information package that relates to it.

They also materialize a genuine "data-prior-to-system" approach: these information packages allow us to set system parameters with machine actionable files. For instance, the system can check the conformity of image files with a specific profile of TIFF used at BnF (TIFF 6.0, 24 bits, 300 dpi resolution, BnF watermarking, etc.) each time a package with files whose MIME type is identified as image/tiff is ingested. In this way, data defines and configures processes, not the other way around. This enhances control of the system processes by users that are not IT specialists.

Last but not least, the reference information packages include a sample file or the source code of the tool, with human readable documentation about the format, in order to meet the needs of digital curators and preservation experts. Every aspect of the system functionalities that has an impact on librarianship is documented in SPAR.

3. WORKING THE DATA INTO THE SYSTEM: THE DATA-MANAGEMENT MODULE IN SPAR'S ARCHITECTURE

Having defined the types of data that go into SPAR, we will examine how they are processed and used by the system — to the extent that certain types of ingested data actually determine the settings of the system.

3.1. A modular implementation of the OAIS

From its early stages of inception, SPAR was to be a modular system: in order to allow easier integration of new technology, each main function had to be able to be improved at its own pace. Thus the system was divided into modules following the OAIS functional model entities: Ingest, Data Management, Archival Storage, Access, Administration, and Preservation Planning, the last one to be developed at a later date. They form SPAR's "core".

Additional modules which do not have a direct equivalent in the OAIS functional model have been designed, such as a Rights Management module, which is not yet implemented, or Pre-Ingest modules for each specific ensemble of similar material. The Pre-Ingest phase is meant to harmonize the different digital documents into a SIP that is SPAR-compliant and can be processed in the rest of the system in a generic way.

In this environment, Data Management could be considered as the inner sanctum of the system, along with Storage. It centralizes all the existing data in the system according to a unified data model, making it accessible through the same interface. See Figure 1 below.

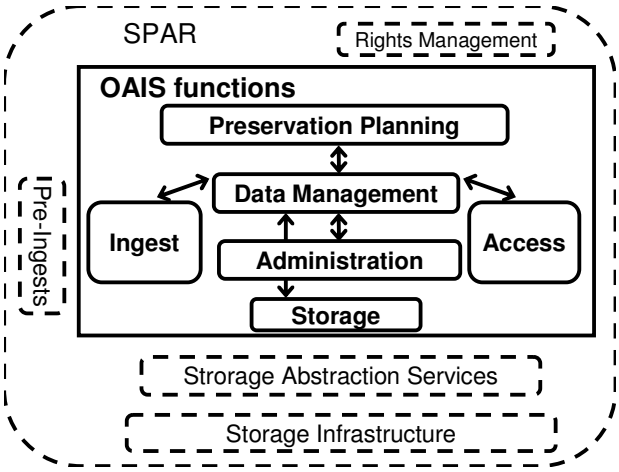

Figure 1. Data Management within SPAR's modular architecture.

3.2. The Data Management module as SPAR's information hub

3.2.1. Data flows between modules

The Data Management module, or DM, is not directly accessible by a human user: every interaction with it is mediated by another module, be it Ingest, Storage, Access or Administration.

All of these interactions have been designed from use cases developed during the specification stage of SPAR. Most of the use cases involve more than two modules, but DM's role in all of them can be viewed as an information hub, managing the metadata flow. All these interactions use RESTful Web services technologies that are compliant with our modular design.

Data Management intervenes in two stages of the ingest process: first during the creation of a SIP, when the latter's characteristics are audited to check their conformity with the channel requirements, stored in the Data Management module, then at the end of the ingest process when the metadata contained in an AIP are recorded into DM.

The Storage module interacts with DM to query reference data and make sure storage requirements for an AIP are met.

Information exchange between the Data Management and Access modules is maybe the most important one

for curators and IT staff to achieve their collection management goals, since any retrieval of data for use out of the system is mediated by Access, whether the data is simply identifiers or more structured information about the system or the packages. Access also needs information from DM in order to provide DIPs to the users.

Data Management's abilities to sift and reorder information are naturally used by the Administration entity in the daily toil of the system, and should assist the future Preservation Planning in preparing migrations and other preservation actions.

3.2.2. Setting parameters with data

Data Management's role as a central nervous system of SPAR can be illustrated with the example of one particular type of data: the Service Level Agreements (SLAs) contained in channel reference packages.

As seen in paragraph 2.2, channels are defined for a particular set of homogeneous digital material which requires the same services from the Archive. The producers of these digital documents and the Archive write down the exact nature of their commitments to one another in a human-readable agreement, which is transcribed in a machine-actionable set of SLAs, written in XML according to an in-house schema. The exact equivalence of human- and machine-readable SLAs guarantees the user communities that the services agreed upon with the Archive are actually implemented as such. These SLAs, along with schematrons to validate the specific METS profiles used in the channel, form a channel reference package.

For each channel, there are three SLAs: one for ingest, one for preservation and one for access issues. Indeed, the same type of controls, such as file format or number of copies, may be applied very differently in the varying stages of the ingestion / preservation / dissemination process. For instance, for the same package, the SIP and DIP may be stored only once, while the AIP will be stored in several copies.

The SLAs define four types of requirements. Requirements at the channel level include the SLA's validity dates, the opening and closing hours or the maximum unavailability duration of the system, for instance. There are also requirements on packages (minimum and maximum size of package, allowed and denied format types for the channel, AIP retention duration, and so on), on storage (number of copies, presence of encryption, etc.) and on processes, determining how the system's resources can be mobilized by the channel (minimum and maximum number of invocations of a process for a given period and so on). All those requirements are entered into the Data Management module when a channel reference package is ingested, and set system variables.

To see how this data is used in the daily workings of SPAR, and the Data Management module's role in them, we can take the "Ingest a SIP" use case as an example.

Whenever the Ingest module receives notification of a new SIP, it is audited, and its METS manifest is validated using reference data that has been put into DM, notably information from the channel package: which users are authorized to submit packages in this channel, or what the METS profile for the SIPs of this channel is.

Then, using DM's capabilities as an index of all the packages in SPAR, the system checks the SIP's identifiers against those of the AIPs already stored to determine whether the SIP is a brand new package or an update, and if so, what type of update.

The SIP's characteristics are checked against the channel service level agreements in DM, such as the maximum size or the number of objects allowed in the package.

The files are individually identified, characterized and validated using tools documented in DM through reference packages. The result is compared with the list of formats accepted in the channel, listed in the SLAs. The behavior of the system if the criteria of the SLAs are not met (rejection of the package or mere warning to administrators) is also specified in the SLAs stored in DM.

Finally, a unique identifier is created for the package, and all the new metadata are added to the package's METS manifest, before the AIP is stored in the Storage module. At the same time, the information present in the METS file is added to the Data Management module.

3.3. The inner workings of the DM module

3.3.1. Different repositories for different needs

The Data Management module as a whole is a data repository, but it is actually divided into a Reference documents repository and a Metadata repository. The Reference documents repository contains documents used in controlling the validity of data and metadata, such as XML schemas and schematrons. The Metadata repository contains representation information and preservation description information that has been transformed from its submitted XML encoding into RDF/XML when inserted into the Data Management module.

The choice of RDF triple stores was made following an extensive risk analysis based on the desired features of the main metadata repositories in SPAR (see 4.1.1). Resource Description Framework has a very generic and versatile data model, where the information is expressed in triples, following the syntax subject/predicate/object. It came ahead in the analysis due to its very flexible query language, SPARQL, and its good performances in mapping from the existing XML metadata and in reversibility. The benefits and challenges of that choice will be further examined in part 4.

The Metadata repository is actually composed of three separate RDF repositories. Metadata from all the AIPs in SPAR goes into the Complete metadata repository, where it is available for complex queries by

the digital collection curators. From the complete metadata, a lighter, faster Selected metadata repository is extracted, to fulfil the metadata needs of the modules of SPAR themselves. Additionally, all the content of the reference packages, which is heavily used in the workings of the system, has its own Reference data repository. See Figure 2 below.

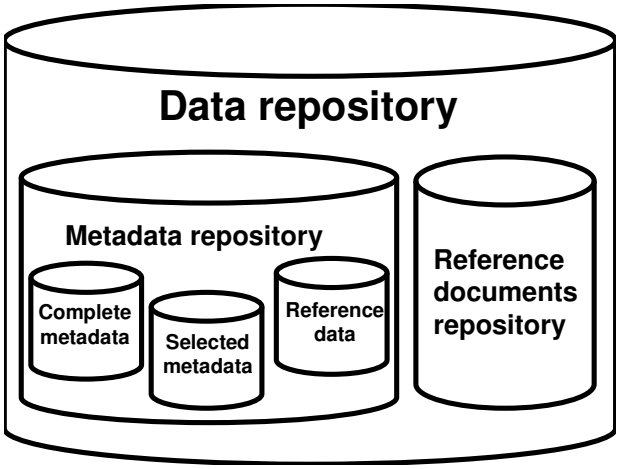

Figure 2. Data repositories in the Data Management module.

3.3.2. Making changes in the data model possible

In order to be useful, data repositories have to be up-to-date. Mechanisms are implemented in order to reconstruct the metadata repositories when new packages are added and updated. However, given the amount of metadata and reference information in the SPAR system, we had to accept compromises, and devise fail-safes.

The Complete metadata repository is not an exact one-to-one transposition of each metadata entry in the METS files of each package: some of the information is not expressed; some of it has been aggregated. For instance, the format of each individual file is not expressed in the triple store; instead the types of format a fileGrp contains will be listed for each fileGrp.

The choice of what information to keep in the RDF triple stores was based on a clear principle: it should be information that the system's users may need to query in order to select and retrieve packages according to an identified professional use. Once the packages have been retrieved and accessed via RDF requests, more detailed actions can be taken after examining the METS files themselves. Detailed examples are provided in 4.2.2.

Of course, some of the information we need in order to identify certain AIPs may have been overlooked in our initial METS to RDF mapping, and our activities will probably change over time (See Bermès and Poupeau [3]). Moreover, the data model may evolve to include new types of information we hadn't foreseen. Thus, the METS files are archived independently, and may be the basis of a planned reconstruction of the Complete metadata repository.

4. THE RDF DATA MODEL: HOW TO SPEAK THE SPAR LANGUAGE

4.1. Principles and methodology

The risk analysis that was performed when the Data Management module was designed pointed to RDF triple stores as the least risky choice of four, when compared to relational databases, XML databases and search engines. Three families of risks were evaluated:

- risks in setting up the technology in SPAR, which included integrating the technology into the system's modules, and mapping the data from METS to the chosen solution;
- risks in managing the metadata: RDF scored very well in querying capabilities, but had higher risks regarding update features;
- risks in maintaining the technology over time: RDF's handling of data models was a plus, but the technology was still new at the time (see 4.2.2 and 4.2.3).

The choice of RDF in itself is far from enough to build an efficient data model. No domain specific ontology, that is, RDF vocabulary, existed in digital preservation when we started building the data model, so we had to build it from scratch according to the following principles.

4.1.1. Using the OAIS information model

While building our RDF data model, we had the same guidelines as when implementing METS: genericity, interoperability, therefore better maintenance and durability.

Since RDF aims at describing things in a self-declarative fashion, using RDF requires the implementation of a domain specific terminology. In order to structure our own information model, we naturally turned to the OAIS information model, which was at an abstract level, thus generic, and had a very strictly standardized, documented and hierarchized terminology of concepts, which favored interoperability.

We built an ontology per OAIS information type: representation, structure, fixity, provenance and context. An additional class was built, agent, since it was a very well-identified domain by itself. It related as much to context information as to provenance information, and matched an existing PREMIS entity.

4.1.2. Reusing existing ontologies

One of the great features of RDF is its modularity: parts of existing ontologies, such as properties and classes, can be integrated into other ontologies. In reusing those parts that are already well-modeled and widely used, SPAR's data model gains a better conformity to existing standards, and we gained time to concentrate on developing our specific classes and properties. However, we are also bound by the intentions of these other ontologies' creators and should not bend these existing rules.

We reused Dublin Core properties[1] for descriptive information, in our reference ontology; OAI-ORE[2] and its concept of aggregation in our structure ontology, to describe relationships between granularity levels; FOAF[3] for agent information and more specifically DOAP[4] for software agents; and so on.

4.1.3. Naming resources with URIs: info:bnf and ARK.

In RDF, resources and properties must be named with URIs. BnF already implements the ARK (Archival Resource Key) URI scheme for its digital material and metadata records. Its open source, non-proprietary nature and maintenance by a public institution (California Digital Library) made it an ideal scheme to use in a digital preservation context as well.

ARK is particularly suited to identify concrete objects, since it can point to parts or specific views of the document with "qualifiers"[5]. For instance,

- ark:/12148/bpt6k70861t names a AIP containing a digitized edition of Charles Baudelaire's 1857 *Les Fleurs du Mal*;
- ark:/12148/bpt6k70658c.version0 names the initial version of this digital document;
- ark:/12148/bpt6k70658c/f5.version0 names the 5th page of this document;
- ark:/12148/bpt6k70658c/f5/master.version0 and ark:/12148/bpt6k70658c/f5/ocr.version0 respectively name the image and ocr files for this page.

Thus, ARK is the way we name actual AIPs, or parts or them, to say something about them in RDF.

But ARK is not suitable for naming abstract information in SPAR, that is, specific properties and classes of our ontologies. ARK names have to be opaque whereas the self-declarative philosophy of the Semantic Web, and usability issues of course, require significant URIs.

To this purpose, SPAR uses the info:uri scheme. For instance info:bnf/spar/provenance# is the URI naming the representation ontology in the system, and info:bnf/spar/provenance#digitization names the abstract event "digitization".

4.2. The result: ontologies and access to data

4.2.1. An ontology: provenance

The provenance ontology in SPAR is very close to the PREMIS data model and shows many features of RDF, as Figure 3 below demonstrates.

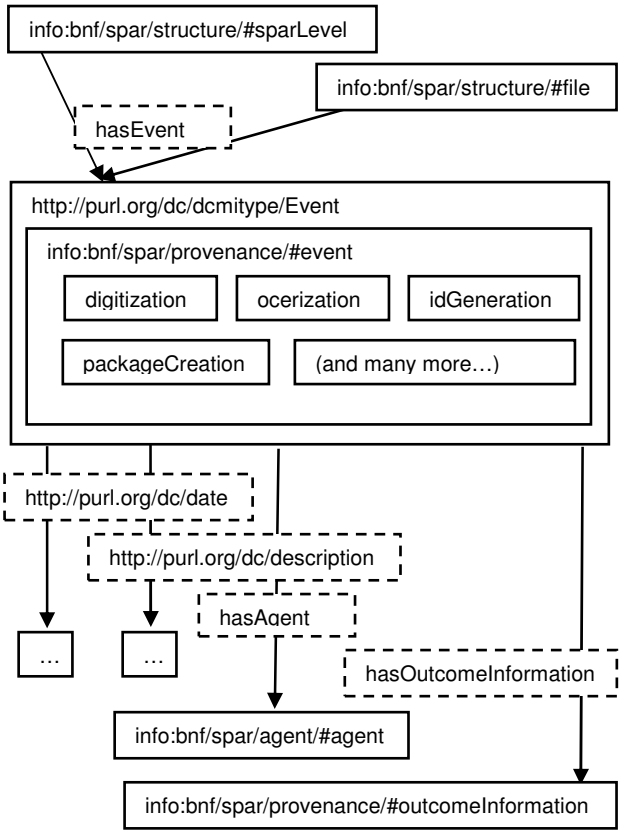

Figure 3. A simplified view of the provenance ontology

As in PREMIS, each event is viewed as an entity relating on one hand to an object, which can be at various granularity levels, and initiated on the other hand by an agent, be it human or software.

Each particular <premis:eventType> in SPAR's METS implementation in XML is modeled as a distinct class with "event" as a common superclass in RDF. For example, the digitization eventType becomes the class info:bnf/spar/provenance#digitization, being a subclass of info:bnf/spar/provenance#event.

Existing properties are reused to express some premis:event elements, as http://purl.org/dc/elements/1.1/description for eventDetail, viewed as the description of an event, or http://purl.org/dc/elements/1.1/date for eventDateTime.

4.2.2. Access to data

The advantages of RDF listed above are particularly valuable when it comes to data retrieval issues.

Data is controlled, thus access is controlled : the same concepts and things always have the same name, that is the same URIs. Queries are precise because they go through controlled access points. And, contrary to what happens with relational databases technologies, it is not necessary to know the names of the categories of data in advance to formulate a query: they can be deduced from the way the data is structured, by successive queries.

Moreover, RDF's query language, SPARQL, is independent of the way the data is actually written

[1] http://www.purl.org/dc/elements/1.1/ for simple Dublin Core and http://www.purl.org/dc/terms/ for qualified Dublin Core.
[2] http://www.openarchives.org/ore/1.0/rdfxml/
[3] http://xmlns.com/foaf/spec/
[4] https://usefulinc.com/doap/
[5] That is, suffixes beginning with "." or "/".

down: the Data Management module uses RDF/XML, but the queries use the abstract way the data are modeled in the subject/predicate/object fashion. Although SPARQL has its own set of rules, compared to other query languages, it follows a common language pattern and is thus more intuitive. And simple query sentences can be assembled to create complex queries.

Here are some examples of queries we can formulate about material from the digitized books and still images collections:

- Which package has pages flagged as containing a table of contents, but no table of contents file in XML, which would allow dynamic navigation in the document? Answering this question helps plan retrospective creation of structured tables of contents.
- How many packages were ingested in SPAR the last month, with their number of files, the formats and the quality rate of the OCR? This traditional question shows that data also helps administrators monitor the system.
- Which packages in our digitization channel have invalid HTML table of content files? Invalid HTML doesn't necessarily impede access to the document, but is certainly harder to preserve; such a query helps preservation experts plan invalid HTML files regeneration.

4.3. Challenges and uncertainties

Even though the BnF sees many advantages in the use of RDF to manage the data in its digital trusted repository, there are many uncertainties and problems attached to adopting a relatively new technology, mainly performance, maintainability and training issues.

4.3.1. Too much information?

RDF remains a recent technology with the weaknesses inherent to its newness, which we faced when implementing Data Management. First, compared to other technologies, few software providers are available for RDF triple stores; only Virtuoso suited our needs in terms of data volume and performance, and yet its implementation required a great amount of tuning and optimization. Its performances are also slower for the moment than those of traditional relational databases. Even though it may not be a foremost issue in a preservation perspective, quick response times give valuable comfort to digital curators.

This problem is exacerbated by one of the principles presiding to SPAR's creation: to use as many open source programs as possible, in order to reduce specific developments, benefit from other communities' maintenance, and enhance financial viability.

However, tests conducted in 2008 showed that our implementation of a Virtuoso Open Source triple store reached its limits when the data volume nears 2 billion triples — although it should be noted that the performances of RDF technologies are improving steadily. 2 billions may seem like a high maximum, but, considering the first channel of documents to be ingested in SPAR already includes 1 million packages with an average of 200 files and at least 5 types of metadata expressed in METS at file level[1], this amounts to 1 billion triples for basic file-level information in one single channel.

Hence the distinction between information useful to identify and access the packages, which is indexed in RDF, and information only needed once the digital documents are retrieved mentioned in 3.3.2. It enabled us to reduce considerably the amount of data indexed in the Data Management module the first channel to enter SPAR in order to gain computing power, while maintaining usability.

4.3.2. New technologies, new skills

Using RDF had other immediate drawbacks for the staff of the BnF, be it on the IT or on the librarian side.

On the IT side, Semantic Web technologies were previously unused at BnF, and require training, first for the digital preservation team, then for their collaborators. Day-to-day monitoring of the Data Management module is also more difficult, since there is little peer support or experience feedback yet.

On the librarian side, training issues are even greater, since SPAR, as a digital collection preservation and management tool, is not only intended to be used by digital preservation experts, but also by producers of data-objects and collection curators (see Bermès and Fauduet [2]). They have to understand SPAR's data model in order to express their information needs. Digital preservation experts and digital data producers may have to act as an intermediate in the beginning, but ideally, everyone dealing with digital collections should be able to get the information they need directly from Data Management, which implies learning how to query it with SPARQL.

Moreover, the lack of well-established best practices in RDF modeling for digital preservation forced us to build SPAR's data model and the ontologies "on the fly", using common sense and professional experience in data modeling.

But all these are difficulties in the short or medium term. In a long-term perspective, RDF has real organizational advantages, as it allows the separation of technical/IT issues from data/librarian ones. As complex as RDF and SPARQL can seem to be in the beginning (but is MARC any easier?), they give librarians a better control of their data, which also equates, in a data-first approach, to a better control of the system processes.

Ultimately, we hope that SPAR's data model, and its use of RDF technologies, will allow all BnF's staff dealing with digital collections preservation and curation

[1] That is, the MIME type of the file, its size, checksum, checksum type, and… the information that each file is a file.

to speak a common language that will adapt to different missions and different time constraints.
Every person in interaction with the Archive will have to refer to the same data model, using the same request language, whether they are planning long-term preservation actions such as migrations; have short-term decisions to make, requesting a new ocerization on certain documents for instance; or need the day's latest statistics. And eventually, all these users will have to define the necessary evolutions of the data model together. This could be the best way to integrate SPAR into the large and diverse ecosystem of the Bibliothèque nationale de France's activities; data-first, the rest should follow.

5. REFERENCES

[1] Bermès, E et al. "Digital preservation at the National Library of France: a technical and organizational overview", *World Library And Information Congress: 74th IFLA General Conference And Council*, 2008. Online at http://archive.ifla.org/IV/ifla74/papers/084-Bermes_Carbone_Ledoux_Lupovici-en.pdf [last consultation 2010-05-04].

[2] Bermès, E. and Fauduet, L. "The Human Face of Digital Preservation: Organizational and Staff Challenges, and Initiatives at the Bibliothèque nationale de France", *Proceedings of iPRES 2009: the Sixth International Conference on Preservation of Digital Objects*, 2009. Online at http://escholarship.org/uc/item/6bt4v3zs [last consultation 2010-04-20].

[3] Bermès, E and Poupeau, G. "Semantic Web technologies for digital preservation: the SPAR project", *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008)*, 2008. Online at http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-401/iswc2008pd_submission_14.pdf [last consultation 2010-05-04].

[4] Guenther, R. "Battle of the Buzzwords: Flexibility vs. Interoperability When Implementing PREMIS in METS", *D-LIB Magazine*, July/August 2008. Online at http://www.dlib.org/dlib/july08/guenther/07guenther.html [last consultation 2010-04-20].

[5] Martin, F. "Dynamic management of digital rights for long-term preservation: the expert system approach", *Proceedings of iPRES 2004: the Fourth International Conference on Preservation of Digital Objects*. Available online at http://ipres.las.ac.cn/pdf/Martin_presentation_Martin.pdf [last consultation 2010-05-04].