The New Numdam Platform

Thierry Bouche, Olivier Labbe

The New Numdam platform

Thierry Bouche[1] & Olivier Labbe[1]

[1]Mathdoc (UMS 5638)
Univ. Grenoble Alpes, Mathdoc, F-38000 Grenoble, France
olivier.labbe@univ-grenoble-alpes.fr
thierry.bouche@univ-grenoble-alpes.fr

This paper is dedicated to the memory of Claude Goutorbe (1952-2016).

Abstract. The Numdam French digital mathematics library has now been in operation for more than 15 years with no major upgrade. It holds more than 57000 documents either scanned or born digital (spanning over one million pages). The information system has been recently completely redesigned. In this article, we present the new Numdam ecosystem. A metadata factory is used to store metadata from a variety of sources, normalize it under JATS (articles) or BITS (books) XML formats, and enhance it through manual editing or automated agents (tagging math formulas, matching to external databases and interlinking, etc.). The data model supports the main types of documents currently expected to populate the DML: journals, seminars, conference proceedings, multivolume works, books, book parts, doctoral theses. All documents are in collections that can belong to one or more corpus. The workflow has been simplified and allows easy deployment on test and production web sites. A platform holds all the data in one place, can generate multiple web sites, each with a different view on the data, and provides an OAI-PMH server to the outside world. Finally, the article presents future plans to create a DML-ready platform based on the new Numdam platform.

1 Introduction

The Numdam programme [4] was launched at the very beginning of this century. It started at Cellule Mathdoc as a pilot digitization project of 5 serials with the goal to provide digital preservation and wide access to our mathematical heritage. The paper collections were borrowed from publishers or libraries, the digitization itself was outsourced. At that time, it was also considered to outsource the online posting as Mathdoc felt it had little resources to manage it. However, while the collections were under development, Claude Goutorbe tested the possibility to build a web site on top of Numdam metadata based on the EDBM technology that he had developed for the first web site of the Zentralblatt MATH database. EDBM (supposedly an acronym for *European database manager*) is basically an indexing engine, with a library for searching, extended progressively with services written in Python. As it appeared very rapidly that this was going to work pretty well, and enabled us to keep access free to the Numdam content by avoiding external costs, the Numdam web site was open to the public back in 2002, based on an evolution of EDBM.

Numdam has evolved to become the French digital mathematics library, with more than 1 million pages acquired from a number of sources (our own digitization amounts to about 70 % of the content, partnering publishers provide born digital files and metadata for the recent articles). Last year, the web site and associated tools such as the OAI-PMH server were still based on the 15 years old EDBM. Also, a custom internal XML format was used to store metadata, meant to capture the metadata we collected for the first digitized series, with some extensions added along the way, but still designed exclusively for articles published in journals. This became more and more troublesome as we were adding different content types such as books or theses.

For instance, when Numdam participated in the EuDML project [5], we had to convert our metadata to the EuDML metadata schema [6] based on the NLM Journal Archiving and Interchange Tag Suite (JATS), and set-up a separate OAI-PMH server [19] in order to expose Numdam content to the EuDML harvester.

Work on a full redesign of the Numdam hosting and dissemination platform has started a couple of years ago, initially led by the late Claude Goutorbe, now by the second author. An entirely new workflow has been built, with components dedicated to specific tasks and communication protocols between them. These components have now been released and are used in production. The Numdam web site [18] is running the new software since February 2017.

This article presents the new Numdam ecosystem: its architecture, the concept of a platform and its advantages to enable the construction of a larger virtual library, and the software engineering practices applied to this project to improve quality and prepare future changes.

2 Numdam architecture

The overall Numdam architecture is presented in Figure 1.

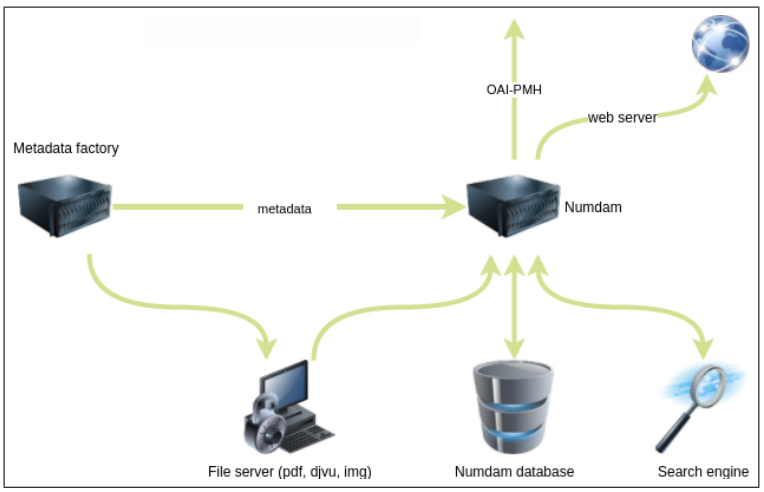


Fig. 1. Numdam platform architecture.

2.1 The Metadata Factory

A separate application, the metadata factory, shown on Figure 2, imports and enhances metadata in the overall system. Documents from a variety of sources (digitized by Mathdoc, by other partners, or born digital documents) are all ingested in the metadata factory. A web interface allows manual modification of any metadata snippet (see figure 3). A number of services that enhance metadata can also be launched selectively on a range of items. The metadata factory software is based on eXistdb, which is a native XML database system [7], with editing templates adapted to our use of JATS and BITS.

This is where we perform the Extract Transform Load (ETL) steps (see, e.g. [22]). In our case, this means that for each ingested item, the XML is either loaded without change if it is tagged according to the new internal format (based on JATS for journal articles [16], and BITS for books [17]), or transformed on the fly to that format if it is tagged under our legacy custom DTD. For metadata coming from publishers, we typically perform transformations to JATS before ingestion.

An anti-pattern is a common structure or set of actions that initially looks like an appropriate and effective solution, but leads you eventually into trouble [15]. As noted by Ken Farmer in his list of ETL anti-patterns [8], a single program must not be used to perform all the steps. Therefore, we developed and use separate automated agents to enhance metadata:

- we use a spell checker in order to find OCR or typing errors in metadata such as titles or author names.
- we have a routine to correct some usual orthotypographical mistakes such as abusive capitalization or wrong spacing.
- we have separate routines to look up external databases such as EuDML, MathSciNet or zbMATH in order to provide deep interlinking for our articles or citations,
- we generate LaTeX driver files that produce the PDF full texts with cover pages.
- as we input all mathematical formulas as LaTeX, they are converted on the fly to MathML and stored in the XML file next to the TeX code,
- we also have a number of programs looking up our metadata for possible problems or errors (such as a relation from errata without backward relation. . .).

In this way, all the ingestion, manual or automated enhancements are managed in the metadata factory, which holds our reference metadata, which is fed to the public web site through a REST API.

2.2 The Numdam web site

The Numdam web site is based on a standard architecture:

- metadata are stored in a relational database,
- the Solr search engine is used to search keywords and provide facets in the search results,
- a file server returns the documents to the user (PDF, DjVu). A protection mechanism prevents users from downloading the full text of the most recent articles for journals following a moving wall principle,

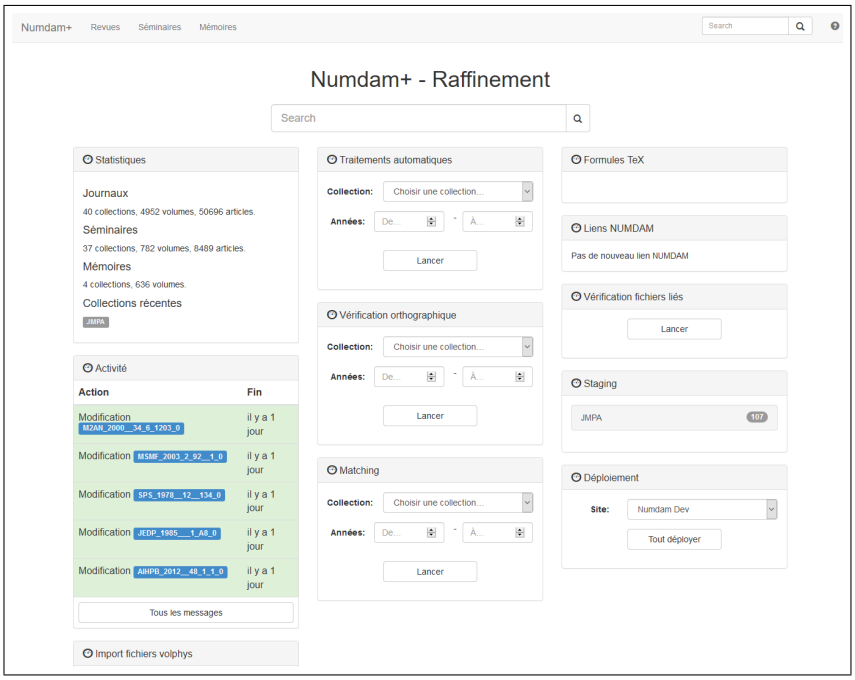


Fig. 2. The metadata factory dashboard.

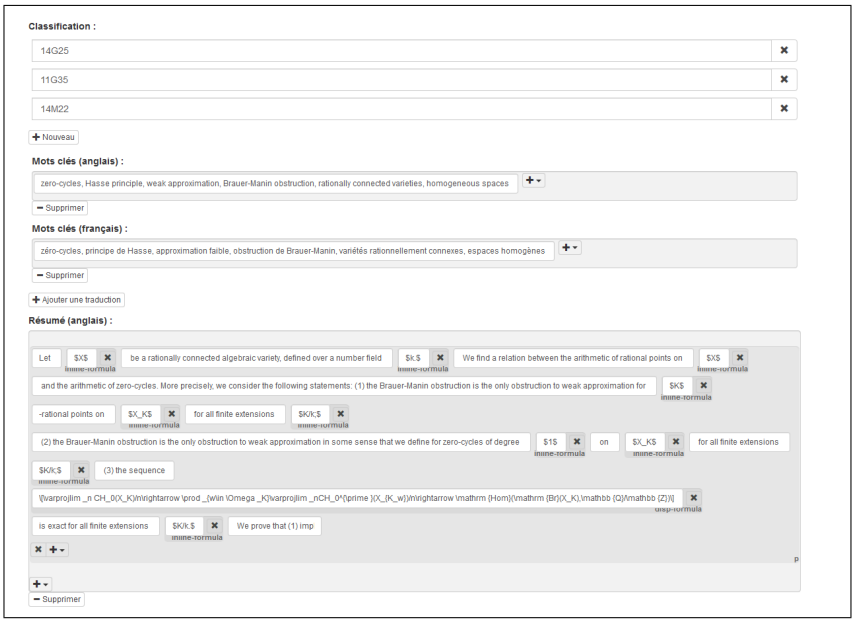


Fig. 3. The metadata factory: editing XML metadata.

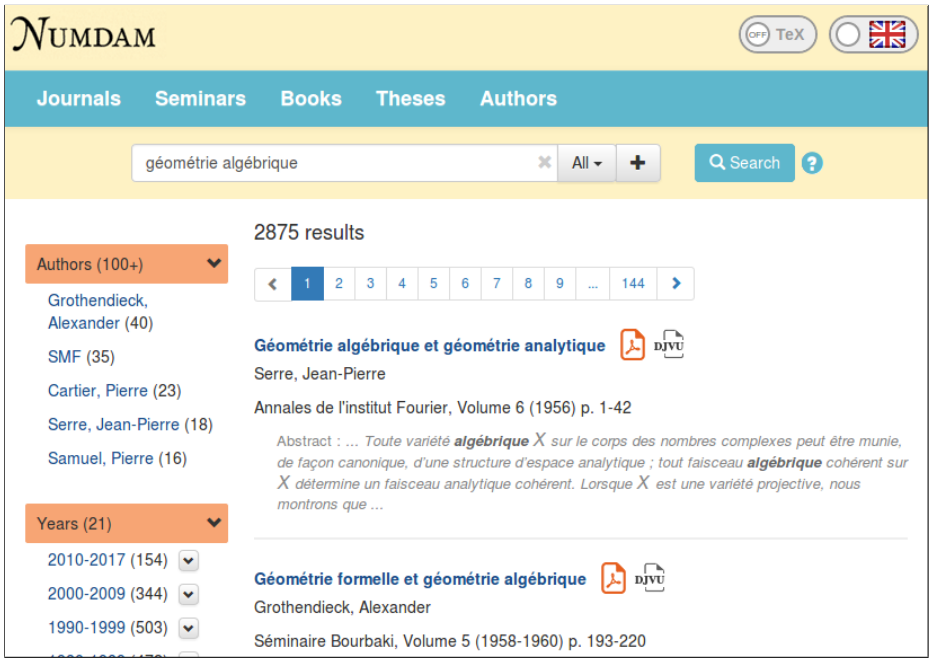
- a web application, developed using the popular framework Django, displays the Numdam web pages in a browser interface,
- an OAI-PMH server exports metadata in multiple formats: the mandatory simple oai-dc, eudlm-article2 and eudml-book2 EuDML formats for much more details, and gallica (which exposes journal-level metadata in oai-dc).

The legacy Numdam (see Figure 4) was based on a custom made search engine in the past. All metadata was stored in a PostgreSQL database with a structure modelled on our legacy (journal-only) DTD. All dynamical pages were produced on the fly as a result of a search query: The search and browsing interfaces, as well as the items' short and long record display pages were generated as a response to a search query. The data returned by the database was assembled (and transformed with XSLT) in order to build the page content. We didn't have a metadata factory but metadata was prepared using various scripts and manual editing yielding a collection of XML files that were ingested and indexed by EDBM. EDBM also created special indexes for features that were not implemented in the XML files (such as reverse citations, links to citing items from a cited item, etc.). We also had a special trouble coming from the fact that the content managing system SPIP [24] was used for the static web pages at `www.numdam.org`. The dynamic part of the web site had to mimick SPIP pages and menus, with hard coded links to SPIP articles. As SPIP and EDBM didn't manage some features the same way (like, for instance, bilingualism: cookies in the cas of SPIP, parameters in the case of EDBM), this could lead to a quite uneven user experience such as launching a search query in the English interface and ending up in a page written in French...

The new search engine is based on the Lucene search library. Communication with the Lucene engine is based on standards like XML, JSON and HTTP. Solr follows the NoSQL (Not only SQL) movement : the data stores are non-relational and therefore do not require fixed table schemas. They avoid join operations to retrieve documents, improving the search operation for complex queries. Solr brings additional benefits, such as faceting, spell checking, similar item search, hit highlighting and free text search. XSLT (EXtensible Stylesheet Language) was used to transform the model into XHTML, which was fine when user interactions were few and pages did not react to the environment, like the size of the browser window. The Model-View-Controller (MVC) design pattern [10] is now applied to separate the view from the user interactions, which are more important today with search facets used to filter search results. Graphic designers can now focus on the ergonomics of the web site, and can provide an adaptive view that reacts to the size of the window.

For a user, the Numdam interface looks like a common one for a digital library: menus allow navigation across collections, search fields let you specify search keywords, and facets let you quickly narrow the search results (see Figure 4). In the past, Numdam objects were built to model a journal. Hacks had to be put in place to support memoirs and doctoral theses. The fact that we now have native support for other document types than journal articles has been used to have a different item page layout (aka landing page) depending on the item type (books or theses, edited books, chapters in edited books...). We are also experimenting with a specific browsing interface for books, ordered as in library shelves by author, year, title.

Fig. 4. The old Numdam web site.



All the content is now organized in three major document types: journal article, single book, and multiple volume book. In Numdam, we have composed our objects into tree structures to represent a part-whole hierarchy, thus following the Composite design pattern in Object-Oriented software [10].

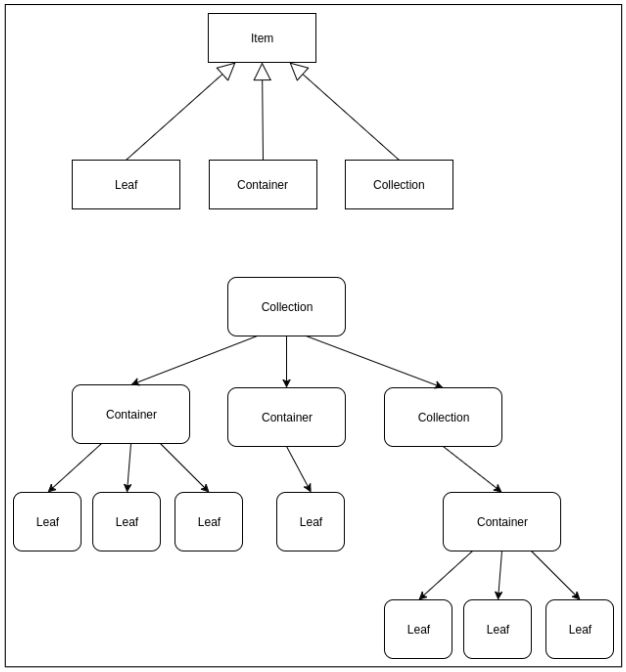


Fig. 5. Part-whole hierarchy.

Figure 5 shows the objects used in our model:
- leaf: journal article or book part,
- container: journal issue, book (including monograph and edited book),
- collection: journal, book series, multivolume work.

As a collection is a composite, it can be embedded in a larger collection. For example, the Séminaire Bourbaki is a collection available in Numdam that happens to have been published in different venues over time (including Springer Lecture Notes in Mathematics, SMF's *Astérisque* series). A subset of this collection is thus part of the *Astérisque* collection, whose other volumes are currently being digitized and will soon be added to Numdam.

2.3 Mathematical formulas

The handling of mathematical formulas has been improved thanks to the new system. In the previous Numdam system, we encoded metadata using an XML DTD that didn't

know anything about mathematical content. As a result, we stored TEX math encoding (with dollars) as text strings (with special XML characters escaped). Thus the math in Numdam, although a reference math library, was not handled as such, and not very much exploitable as math knowledge. We did tag formulas as math and convert most of them to MathML when we generated JATS XML to be ingested in EuDML, but we could not exploit this on our own site!

Now the metadata factory automatically detects LATEX math environments while editing text blocks that may contain it (such as title, abstract, keyword...). The JATS XML superstructure identifying formulas is added, and the MathML alternative to the TEX code is added. Team members without computational skills can now use the metadata factory, edit mathematical formulas and immediately preview the graphical form. Automated tests have been written to verify the validity of the TEX source code in all the articles stored in Numdam. We were able to detect and fix a little less than 100 issues. The improvement in the mathematical formulas handling allowed us to set additional goals and support mathematical formulas in references.

By default, mathematical formulas are displayed in graphical form using MathJax from MathML (see Figure 6). A switch, easily accessible in all page headers, lets you display the TEX source code of the formulas for accessibility or copy-pasting (see Figure 7).

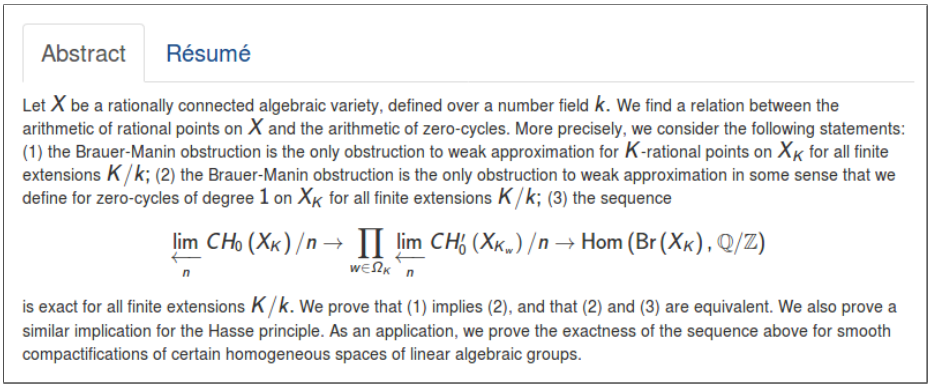


Fig. 6. Mathematical formulas in graphical form.

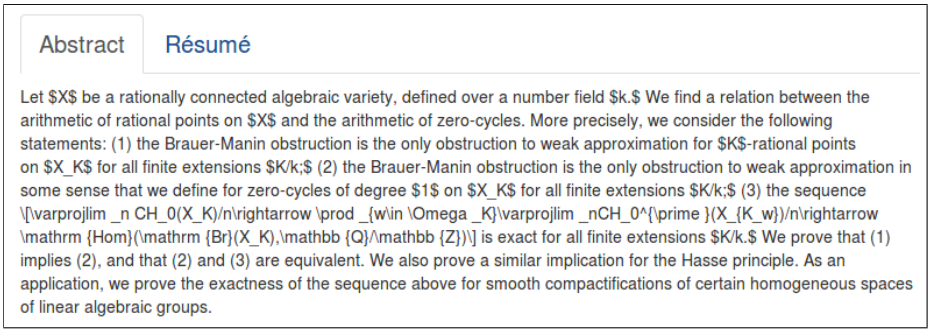


Fig. 7. TEX source code of the formulas.

These changes allow us to be fully embedded in the network of DMLs as we are now able to serve MathML (and, in fact, some other math-specific goodies such as mathematical subject classification codes) and full EuDML compliant metadata through our OAI-PMH server.

3 Numdam platform

The new Numdam has been designed so that the same platform can generate multiple sites, all of them using the same data (PDF, DjVu, images, metadata, search engine). The industry platform definition is used here: "a set of assets organized in a common structure from which a company can efficiently develop and produce a stream of derivative products" [11].

The platform development was initially started in order to provide Mathdoc with a versatile tool for storing and delivering documents that we host like the Numdam collections, as well as some companion collections that will be added to the Numdam web site as "associated libraries".

An important feature of the platform is that it is very easy to update the platform content through its REST API, and the new system supports incremental updates (at the granularity of journal issue or single book), which will simplify a lot the online posting workflow as the (non-technical) people using the metadata factory can check their editing on a test web site until they are happy with it, then switch the validated items to the public web site. This is in strong contrast with the previous EDBM-based system where a large global index had to be recomputed from scratch for each modification.

We are also currently working on an extension to the platform that will replace the current farm of EDBM-powered Cedram web sites (where we have one overall web site with a global search engine, and one web site for each journal with a specific layout and web design, see [3]). As each journal publishes regularly content indepedant from each other, the possibility to add or remove articles while preparing new issues will be of great value to our staff.

Figure 8 shows 3 sites: `numdam.org` and 2 journal's web sites (`aif.cedram.org` and `afst.cedram.org`) that can be based on the same platform. The 3 sites can be hosted on the same physical server, or they can be cloned on separate virtual machines if needed, to improve performance.

An advantage of such an architecture is that bibliographical data can be collated and curated in only one place. Harvesting the metadata gathered by Mathdoc in a bigger virtual library is facilitated. As we have much better quality control on our metadata and native support to serve full math-aware metadata, the Numdam platform helps furthering the DML objective, this time using the DML definition of the "virtual library of all mathematical texts available digitally, both retrodigitized and born digital" [2]. As all document types envisioned to form the DML are natively supported by the platform, it can also serve to run a large DML system for content delivery, with content harvested through OAI-PMH or other means, this is a goal we intend to pursue in a near future.

In order to make the new Numdam platform successful, we had to prepare the change. The first step was to realize that Numdam was not just a software application, but an information system: a "Formal, sociotechnical, organizational system designed

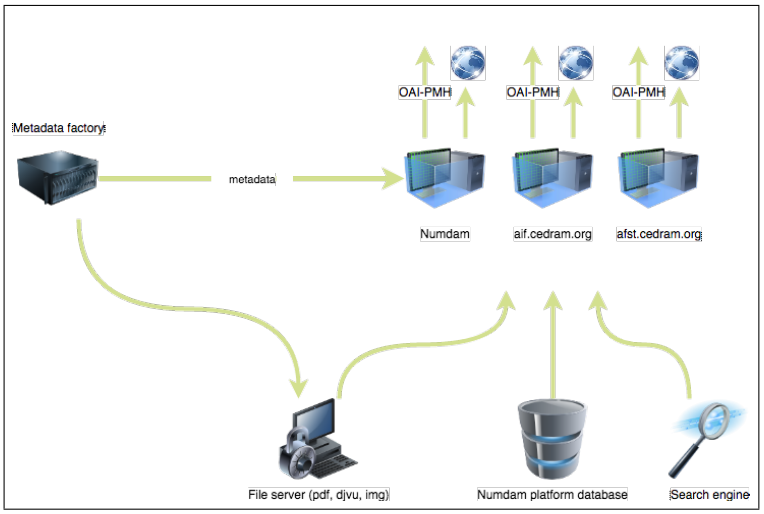


Fig. 8. The New Numdam platform: multisite.

to collect, process, store, and distribute information" [20]. An information system has four components: technology, structure, people and process. Designing an information system requires to consider each component, not just new features or new technologies.

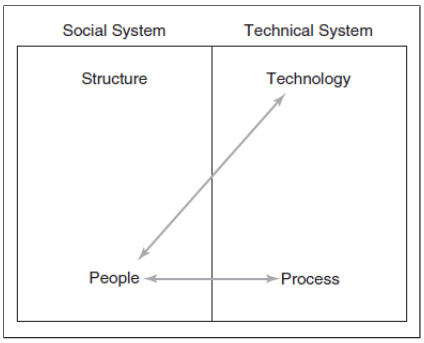


Fig. 9. Second-order change.

In the case of Numdam, a second-order change was anticipated. In a second-order change, the technology and processes evolve, and the people who perform the processes are affected by the change [20] (see Figure 9). The objective was not only to replace the technology to automate some tasks or to use a more modular and modern technology, but also to involve a different set of people. The change was anticipated, and attention was paid to the ease of use of the interface. The metadata factory dashboard displays tasks that can be launched from the web interface, the status of these tasks, and lets you edit any metadata. The interface is now meant to be used by non-technical people. As a result of this second-order change, metadata are now updated more efficiently. They can be detected quickly and corrected before publication, by team members without computational skills.

To ensure that the interface of both the metadata factory and the Numdam web site were intuitive, an agile methodology, based on the SCRUM development process [23] was followed. Features were delivered on a regular basis and presented to the users.

This iterative approach allowed us to implement the most important features first and to collaborate with the users regularly, thus following one of the principles of the Agile manifesto [1] where "customer" collaboration is valued over contract negotiation. Rewriting Numdam was also an opportunity to introduce some of the software engineering best practices. Test-driven development (TDD), where test cases are created before the code is written, helps prevent defects much more efficiently than many other forms of testing [14]. Continuous delivery, in particular deployment automation, reduces the delivery cycle time, and hence gets bug fixes to users faster [12]. This practice enabled us to deploy the source code and all the data (binary files, metadata) to test or production servers easily. In addition, it fostered the development team to submit and deploy the changes frequently, thus iterative development became natural. These software engineering best practices not only improve the software quality, but will also enable us to deliver multiple sites (Numdam, Cedram). The architecture will also make it easier to develop more APIs as the need for them appears, e.g. in the context of an international coordination of a global DML such as envisioned by the International Mathematical Knowledge Trust (IMKT, see [13]).

4 Conclusion

The new Numdam platform, publicly available since early 2017, was designed with a standard architecture (database, search engine, file server, OAI-PMH server, web applications developed with a popular framework). Some of the software engineering best practices were applied which make the source easier to maintain, and the model easier to extend. This platform will enable us to efficiently develop and produce a stream of derivative web sites. The next objective is to support the Cedram family of journals and to add associated collections to Numdam: Gallica-math [9] and Orsay mathematical publications [21] for a start. These collections are currently being posted with *ad hoc* unmaintained applications, and will benefit of much more visibility once integrated into the Numdam platform. Moreover they will be easily contributed to any DML harvester, thus enlarging the rich content already available in, e.g. EuDML.

In this sense, we view the platform as an enabling device to contribute more content to the global DML: We will be able to host in the same system, with the same quality metadata and a versatile format supporting all document types, more collections, including those from institutions that do not have the capacity to properly make them visible and interoperable to the other DML projects (or, in the case of very large generic digital libraries such as Gallica, the Digital Library of the French National Library, that would not allocate resources to a math-only project such as curating mathematical metadata),

Another application of the platform will be to create a DML web site much larger than those already existing. The goal is to create a larger virtual library, by harvesting metadata from other digital libraries, with much more detailed metadata than was available in the mini-DML [2], and a much larger coverage. This will require to set up an OAI-PMH harvester using the REST API in order to populate that instance of the platform.

We do believe that the architecture of the platform will make it easy to build such a virtual library, and to evolve with new APIs and innovative services on top of its holdings in order to keep pace with advances in the DML.

References

1. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for agile software development (2001), http://www.agilemanifesto.org/
2. Bouche, T.: Introducing the mini-DML project. In: Becker, H., Stange, K., Wegner, B. (eds.) New developments in electronic publishing, AMS/SMM Special Session, Houston, May 2004, ECM4 Satellite Conference, Stockholm, June 2004. pp. 19–29. FIZ Karlsruhe / Zentralblatt MATH (2005), http://www.emis.de/proceedings/Stockholm2004/bouche.pdf
3. Bouche, T.: CEDRICS: When CEDRAM meets Tralics. In: Sojka, P. (ed.) Towards Digital Mathematics Library. DML 2008 workshop, Birmingham, UK, July 27th 2008. pp. 3–15. Masaryk University, Brno (2008), http://dml.cz/dmlcz/702544
4. Bouche, T.: Toward a digital mathematics library? In: Borwein, J.M., Rocha, E.M., Rodrigues, J.F. (eds.) Communicating mathematics in the digital era, pp. 47–73. AK Peters (Nov 2008), https://hal.archives-ouvertes.fr/hal-00347682
5. Bouche, T.: Reviving the Free Public Scientific Library in the Digital Age? The EuDML project. In: Kaiser, K., Krantz, S., Wegner, B. (eds.) Topics and Issues in Electronic Publishing, proceedings of the AMS Special Session on Topics and Issues in Electronic Publishing at 2013 Joint Mathematics Meetings, 9–10 January 2013, San Diego, USA. pp. 57–80. FIZ Karlsruhe (2013), http://www.emis.de/proceedings/TIEP2013/05bouche.pdf
6. Bouche, T., Goutorbe, C., Jorda, J.P., Jost, M.: The EuDML metadata schema: Version 1.0. In: Towards a Digital Mathematics Library. Bertinoro, Italy, July 20-21st, 2011. pp. 45–61. Masaryk University Press (2011), http://eudml.org/doc/221288
7. Exist Solutions: eXistdb, an Open Source native XML database, full online documentation at http://exist-db.org
8. Farmer, K.: Data Warehouse ETL for Data Scientists, http://www.ken-far.com/2011/03/data-warehouse-etl-for-data-scientists.html
9. Gallica-Math, a front-end to some mathematical content from Gallica, http://sites.mathdoc.fr/JMPA/
10. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-oriented Software. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1995)
11. Gawer, A., Cusumano, M.A.: Industry platforms and ecosystem innovation. Journal of Product Innovation Management 31(3), 417–433 (2014)
12. Humble, J., Farley, D.: Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation. Addison-Wesley Professional, 1st edn. (2010)
13. Ion, P.D.F., Watt, S.M.: The Global Digital Mathematics Library and the International Mathematical Knowledge Trust. In: Geuvers, H., England, M., Hasan, O., Rabe, F., Teschke, O. (eds.) Intelligent Computer Mathematics 10th International Conference, CICM 2017, Edinburgh, UK, 2017, Proceedings. Springer (2017), this volume
14. Jones, C.: Software Engineering Best Practices. McGraw-Hill, Inc., New York, NY, USA, 1st edn. (2010)

15. Koenig, A.: Patterns and antipatterns. JOOP 8(1), 46–48 (1995)
16. National Center for Biotechnology Information: Journal archiving and interchange tag library, NISO JATS (Aug 2012), full online documentation at http://jats.nlm.nih.gov/1.0/
17. National Center for Biotechnology Information: Book interchange tag suite (BITS) (Aug 2016), full online documentation at https://jats.nlm.nih.gov/extensions/bits/
18. Numdam, The French Digital Mathematical Library, http://www.numdam.org/
19. Open Archives Initiative: Protocol for Metadata Harvesting, documentation at http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm
20. Piccoli, G.: Information Systems for Managers: Texts and Cases. Wiley Publishing, 1st edn. (2007)
21. Publications Mathématiques d'Orsay, http://sites.mathdoc.fr/PMO/
22. SAS: ETL: What it is and why it matters, https://www.sas.com/en_us/insights/data-management/what-is-etl.html
23. Schwaber, K.: Scrum development process. In: Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA. pp. 117–134 (1995)
24. SPIP, a publishing system for the Internet, see https://www.spip.net/en_rubrique25.html

This article is available at Zenodo via https://dx.doi.org/10.5281/zenodo.581405.