CDS homogenisation of metadata from publishers

Grégory Mantelet, Anaïs Oberto, Magali Neuville, Soizick Lesteven, Mark G. Allen

Draft version September 28, 2021
Typeset using LaTeX default style in AASTeX631

CDS homogenisation of metadata from publishers

Grégory Mantelet,[1] Anaïs Oberto,[1] Magali Neuville,[1] Soizick Lesteven,[1] and Mark Allen[1]

[1] *CDS - Centre de Données astronomiques de Strasbourg*
11 rue de l'université
67000 Strasbourg, France

ABSTRACT

The mission of the CDS is to collect, add value to, and distribute the data published in astrophysics journals. CDS is renewing its pipeline dedicated to the analysis of articles published in the main astrophysics journals. This is the entry point for information that is processed for example for the CDS SIMBAD service (Wenger et al. 2000) - for detecting new and known astronomical sources in articles. For efficiency the CDS pipeline needs to download entire volumes or issues of articles for a journal. The text analysis software must behave in the same way regardless the journals. However, publishers provide articles in different formats such as PDF and XML (often with different schema). As such, the CDS pipelines require a pre-processing to convert all these into a single format more suitable for our analysis. Here, we describe the recent efforts to convert all the articles provided by different publishers into a single and homogeneous XML format.

Keywords: Astronomy software (1855) — Astronomy databases (83) — Interdisciplinary astronomy (804) — Astronomy and astrophysics article — SIMBAD

1. INTRODUCTION

The SIMBAD database aims to collect information - such as names and position - about astronomical objects associated with bibliographic references. Thus, thanks to SIMBAD, astronomers can have an overview of all papers published with a study on a specific astronomical object regardless of the name used inside.

To achieve this goal, the documentalists have to analyse every single astronomical and astrophysical article treated at CDS.

They are helped by scripts and home-made internal softwares like DJIN [1] which sole purpose is to parse an entire article in order to locate possible matches for object identifiers. Then documentalists have to confirm, complete or ignore these candidates. An example of an article processing at the different stages is shown in Figure 1.

During the LISA IX Conference [2], we presented a poster introducing an evolution of the way the CDS gets and prepares articles from as many journals as possible in order to help our documentalists in their tremendous daily work.

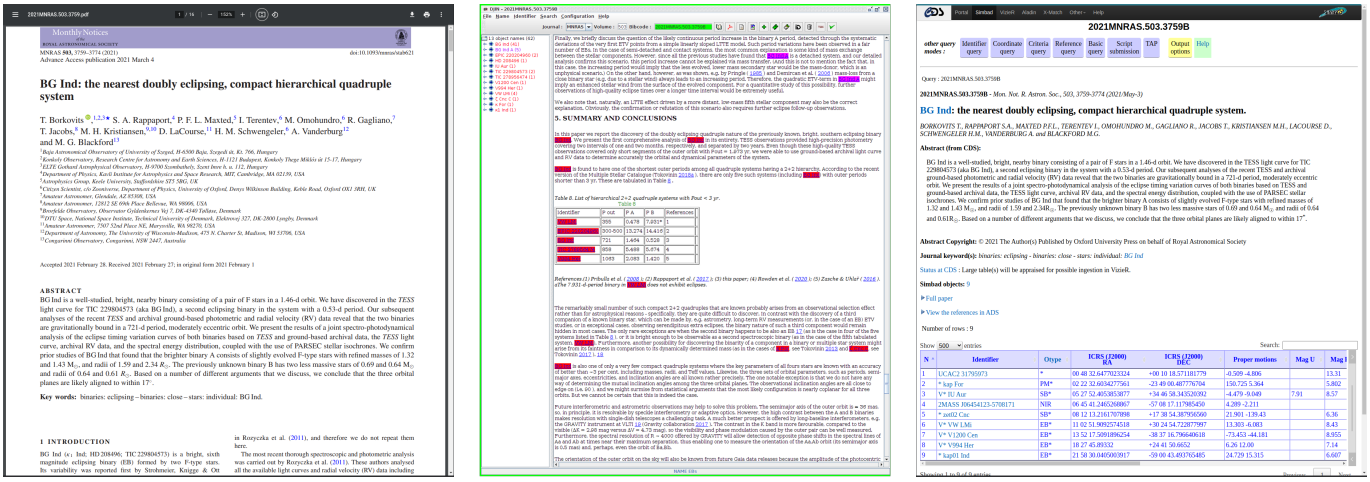
2. CDS' PIPELINE INPUT

Nowadays, depending on the publisher, the CDS gets articles in a different way. In order to prepare them for their analysis by the documentalists, the CDS has to deal with three points:

- the article files,
- the files format,
- the technical way to fetch these files.

Corresponding author: Gregory Mantelet
gregory.mantelet@astro.unistra.fr

[1] Internal CDS Software ; it is an accronym for *Detection in Journals of Identifiers and Names*
[2] https://lisa9.org/



(a) (b) (c)

Figure 1. An article as provided by a publisher (a), then analysed in DJIN (b) and finally after its integration in the SIMBAD Database (c).

2.1. *Article files*

The most important file is obviously the article itself. It can be provided in different formats (see 2.2. Article formats), but generally a PDF version is provided.

Together with the actual article, some publishers provide a file called *TOC* or *Table of Contents*. It contains only metadata about the article, such as the authors, the abstract and the keywords. If not provided, the TOC file can be generated by the SIMBAD team from the article file itself, if its structure embeds enough metadata.

Note that some publishers also give access to other resources such as figures, tables and supplementary materials (e.g. MRT, ZIP archive, videos, ...). All of them are always not as useful for the CDS documentalists using DJIN, but some are essential for other CDS services (e.g. VizieR (Ochsenbein et al. 2000) requires MRT and other tables to feed its database).

2.2. *Article formats*

The actual articles can be provided in numerous formats. The most common formats are PDF and XML. Formats like HTML and plain text are used by less publishers but often for a large number of articles, though not less important for the astronomical community.

2.3. *Fetching articles*

Depending on the publisher, these files can be fetched using different ways:

- sent by email by a publisher to the CDS,
- uploaded by a publisher on a CDS' FTP,
- downloaded by the CDS from a publisher Website,
- downloaded by the CDS with a priviledge access from a publisher's endpoint (e.g. Web service, FTP).

3. CURRENT PIPELINE

3.1. *Description*

Until now, the article ingestion pipeline of the CDS has followed the workflow illustrated in Figure 2.

The first major step of this version of the pipeline is to prepare the bibliographic reference in our SIMBAD database for each article to be processed. This only requires few metadata about the articles, such as the title, authors, keywords, the abstract and the copyright. All these information can be found in the TOC provided with the article or generated by SIMBAD from the article.

Once the bibliographic reference is created, the actual article has to be analysed by the documentalists. The first thing to do is to fetch the actual article, if not already done. Analysing an article is a semi-automatic process. Documentalists use the internal software DJIN to automatically detect object identifiers in function of some predefined patterns (generally built thanks to the existing content of the SIMBAD database and the Dictionary of Nomenclature[3] (Lortet et al. 1994)). The version of DJIN used in this pipeline works *exclusively* on the PDF version of an article.

Then, documentalists have to validate and enrich the automatic result of DJIN. For example, some detected identifiers are not really object identifiers, some are not partially or fully detected, or some are misspelled due to incorrect characters or layout. They also have to determine whether a new identifier should be created or whether an existing object should just be updated. And finally, when an article analysis is complete, documentalists have to prepare and run the update of the SIMBAD database. External files, like tables, are processed later by another team of documentalists and other softwares generally related to VizieR.

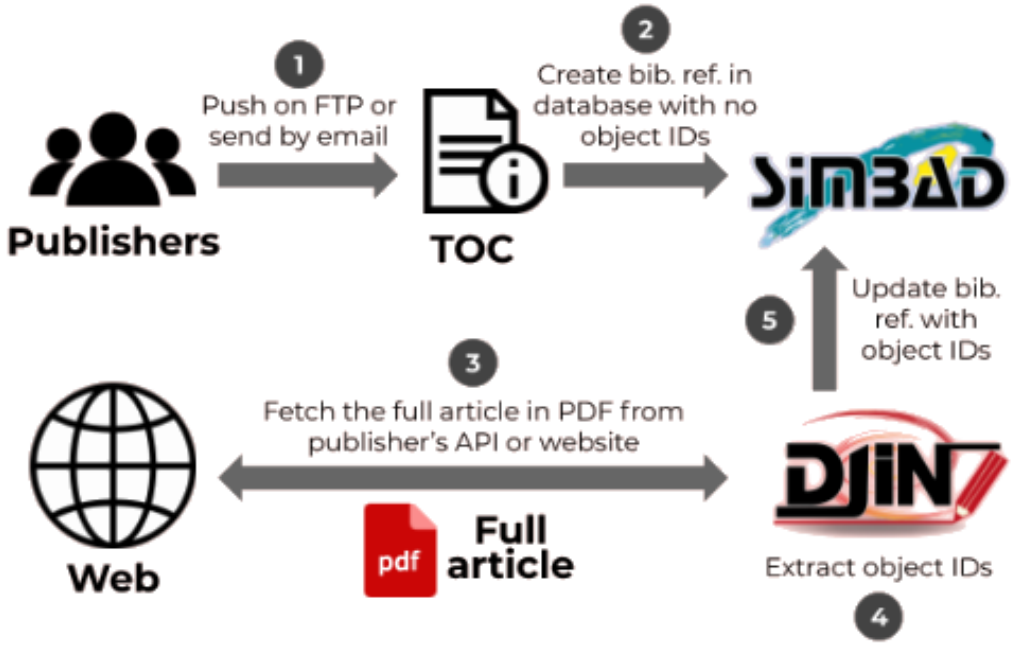


Figure 2. The articles ingestion pipeline used by the CDS until now.

3.2. *Problems*

Various problems are increasingly arising with this pipeline: the file format, the number of input files and the mechanism to get these files.

[3] https://cds.unistra.fr/cgi-bin/Dic-SIMBAD

DJIN encounters more and more difficulties to clearly identify all the different parts of an article (e.g. titles, authors, paragraphs, tables, captions). Besides, depending on the publisher and the publication date, the article layout is not the same ; it changes over time because of updates from the publisher but also from the PDF standard itself. This makes the maintainance of DJIN even more complicated.

The reason for this difficulty is the article format: PDF. This format was originally designed to read documents on screen and to print them. However, it is very difficult to automatically extract information or excerpts of text with a tool (here, DJIN). Doing such extraction requires technics close to image processing. That's why this excercise becomes more and more difficult for DJIN: the document's layout changes over time and from one publisher to another.

Currently the pipeline relies on two files: the TOC and the article itself. As mentionned above, sometimes the TOC can be easily generated from the article. This is only possible if the article format is *reliably machine readable*.

As described, to work, the pipeline needs to get these two files at two different stages (steps 1 and 3 in the Figure 2). However, the way to get these files depends on the publisher and even sometimes on the type of file. Depending on the file to get, the part of the pipeline responsible to get it is not the same. Particularly, DJIN, whose purpose should only be to search for object identifiers in an article, has to deal with the different mechanisms for obtaining the article to be analysed.

What should also be observed is whether the files are downloaded or uploaded. We currently support files uploaded or sent to us (Push method), but also files that we must download (Pull method). After years running this pipeline, it appears that the Pull method is the most interesting one. For some journals, not all articles are about astronomy, and therefore are not analysed at CDS. Downloading articles then gives us more flexibility: we just get the articles we need. It also gives us the possibility to get articles whenever we need them sometime after their publication, as it occasionally happens in case of delay or just to check an old processing. On the contrary, the Push method puts more pressure on our team. Keeping and organizing all articles is not really an issue for us, but processing them as they arrive is. It requires reliable and frequently maintained automatic tools to do that daily otherwise we might loose partial or entire volumes or issues.

Finally, it has to be noted that this pipeline is run only on articles in their final version. Generally, the status of an article depends on the status of its volume or issue. We consider that when the latter is declared as complete, all its articles are finished and therefore ready for processing. However, this rule may not apply to some publishers when articles are still updated after the completeness of their volume/issue. This actually complicates our workflow by forcing us either to wait longer before processing an article since its completeness status or to process an article several times. The first solution is generally preferred.

4. IN PROGRESS PIPELINE

The outcome of the difficulties identified in section 3.2 is that we clearly need to ease the work of DJIN and to simplify the way we get access to the articles.

The most obvious improvement is to change the input format of articles. Instead of PDF, which is not reliable when it comes to text analysis, we would like to use XML format. Indeed, this format is designed for easy processing by machines and thus, is not easily understandable by humans without using a third-party tool to apply a transformation. However, as its tags and structure are free, an XML schema is generally needed for a tool to know how to parse it. Fortunately, there is now an emerging XML schema for scientific publications: JATS [4] (ANSI/NISO Z39.96-2019 2019), which stands for *Journal Article Tag Suite*. Although not all publishers have already adopted it, more and more publishers provide articles in an XML format with an XML schema close to JATS. Of course, even another XML schema is still fine for us as it is still easier to read an article in XML than in PDF. However, having less XML schema to support is more convenient for us as it implies less maintainance and more re-usability.

Note that the HTML format is an official and standardized variant of the XML format but is actually designed to display a document in a Web browser. Consequently, like the PDF, it does not generally come with semantical tags and does not necessarily semantically structure information inside a document. It is then not really an improvement for us, although slightly better than the PDF.

From an XML article, we can easily get all the information provided by the TOC (and eventually generate it if needed) as well as allowing its content to be analysed by DJIN. Choosing the XML format then solves two of our difficulties: the format and the number of files.

[4] Website with documentation and materials: https://jats.nlm.nih.gov/

However, even by adopting the XML format, DJIN must still deal with the different XML schema used by the different publishers. Fortunately, as the XML is easily machine readable, it is possible to transform it into a unique format understandable by any of our internal tools. That way, especially DJIN only has to interpret one input format. As XML is still a convenient format for text analysis, we keep the XML format but we just change the XML schema. The latter will be our own internal XML schema, which, for simplicity, will be called in this article *XCDS* (XML for CDS). Thus, no matter who the publisher is, DJIN will always be able to analyse any article.

Now, to avoid several parts of our pipeline (and especially DJIN) having to fetch this file (and eventually its associated resources), the idea is to create a service only responsible for getting articles. We called this service *BCS*, for *Bibliographic Center Supervisor*. When documentalists want to process a published and complete article, they query the BCS. Depending on the journal, the BCS will choose the appropriate get mechanism, fetch the article and convert it into XCDS. From there, DJIN can get the article and the documentalist can analyse it. The BCS should also allow us to create the bibliographic entry in the SIMBAD database, but this part is still under development.

This new and ongoing pipeline as described here is illustrated in the Figure 3. We can clearly see that there is now a central element: the BCS. It is no longer a linear pipeline in which element has to fetch its own input. Everything about how to get the article files is done by and through the BCS. All tools having to work on an article then work a pivot format - the XCDS - provided by the BCS.

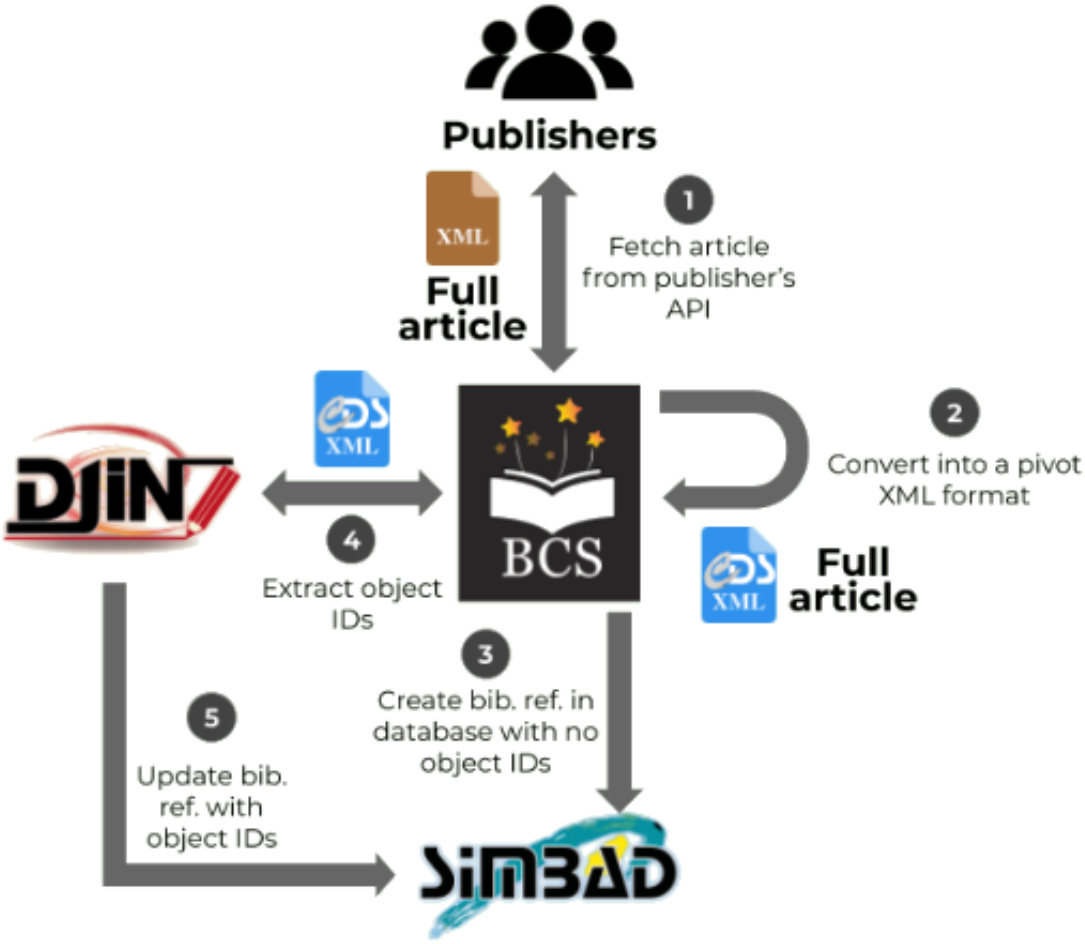


Figure 3. The future articles ingestion pipeline used by the CDS *(still under-development)*.

5. CONCLUSION

The evolution of the CDS' articles ingestion pipeline presented here improves the way the CDS gets and processes astrophysical journals:

- only one file to get from the publishers instead of two - generally the full XML article,
- a centralized service to fetch articles - the BCS,
- a unique article format for DJIN (provided by the BCS) - the so-called XCDS,
- potentially re-usable by other internal CDS services (e.g. VizieR for table processing).

This new pipeline is still an ongoing work. Adjustements will be made if needed when more and more journals will be supported. Currently, this pipeline can fetch articles from roughly 10 journals (from 4 publishers) among the nearly 35 journals now processed at CDS.

In order to support more journals, we will have to deal with some articles not provided in a machine readable format. In such case, there is no choice, we have to work with the PDF version as we did until now. However, as explained previously, parsing a PDF document requires a constant difficult maintainance with a quite disappointing result. Although we would recommand the usage of XML with JATS, any other machine readable format would generally still be better than the PDF format.

Another difficult aspect of our processing workflow is the kind of method used to access articles: Pull or Push method. As explained beforehand, the Pull method is highly preferred by the CDS. Unfortunately, this is not the most used method among the journals we want to process. Publishers have fair technical and organisational reasons for using one solution over another. Hence the BCS, whose mission is to adapt itself, as much as possible, by transparently supporting as many access methods as possible.

In order to make our pipeline more efficient, it would be nice to automate the harvesting process of articles. To make that possible we need a notification system provided by publishers to tell us when a volume or issue is complete. Unfortunately, very few publishers provide such a useful system.

In conclusion, we would like to thank publishers. Although we expressed some desired changes, the current and coming pipelines would not be possible without their existing infrastructure making accessible to us their catalogue of articles. This next version of our pipeline is already a nice start of improvement for our team. It also encourages interesting collaborations with journals and publishers, in the hope that these evolutions will benefit everyone.

REFERENCES

ANSI/NISO Z39.96-2019. 2019, JATS: Journal Article Tag Suite, version 1.2 (National Information Standards Organization, Baltimore, Maryland, U.S.A.). https://www.niso.org/publications/z3996-2019-jats

Lortet, M. C., Borde, S., & Ochsenbein, F. 1994, A&AS, 107, 193

Ochsenbein, F., Bauer, P., & Marcout, J. 2000, A&AS, 143, 23, doi: 10.1051/aas:2000169

Wenger, M., Ochsenbein, F., Egret, D., et al. 2000, A&AS, 143, 9, doi: 10.1051/aas:2000332