IdRef, Paprika and Qualinka. A toolbox for authority data quality and interoperability

Aline Le Provost, Yann Nicolas

Authors
Aline Le Provost
ABES
227 avenue Professeur Jean-Louis Viala
CS 84308
34193 Montpellier Cedex 5
le-provost@abes.fr
https://orcid.org/0000-0001-8823-9048

Yann Nicolas
ABES
227 avenue Professeur Jean-Louis Viala
CS 84308
34193 Montpellier Cedex 5
nicolas@abes.fr
https://orcid.org/0000-0001-5592-7231

Title (English)
IdRef, Paprika and Qualinka. A toolbox for authority data quality and interoperability

Summary (English)
Authority data have always been at the core of library catalogs. Today they are reference data on a wider scale. The former authorities of the "Sudoc" union catalog mutated into "IdRef", a read/write platform of open data and services which seeks to become a national supplier of reliable identifiers for French Universities. To support their dissemination and stick to high quality standards, Paprika and Qualinka have been added to our toolbox, to make easier the massive and secure linking of scientific publications to IdRef authorities.

Keywords (English)
Authority data, Identifiers, Metadata quality, Automation, Curation

Acknowledgments
The authors would like to thank the researchers and engineers of the GraphIK team for their fruitful, friendly and stimulating collaboration: Michel Leclère, Michel Chein, Alain Guttierez, Clément Sipieter and Brett Choquet. We thank ANR for funding the Qualinca project.

Introduction

The Bibliographic Agency for Higher Education (Abes) is a French public institution supervised by the Ministry for Higher Education Research and Innovation.
Abes manages several catalogues and databases with specific contents:

- *Sudoc*, the union catalogue of french university and research libraries ;
- Sudoc-PS, the national bibliography of serials, a subset of the Sudoc ;
- *theses.fr*, the french dissertations search engine ;
- *Calames,* the catalogue of archives and manuscripts held by french universities and research institutions ;
- *Bacon*, the reference database for metadata about electronic resources packages ;

- *IdRef*, the database for authorities.

These databases come with services and tools for librarians to produce, access and reuse data. All these services target at different networks, that span nowadays 1 450 libraries in more than 150 academic and research institutions.
Abes is also involved in national projects such as ISTEX[1] and CollEx-Persée[2] which involve the acquisition, analysis and enrichment of a large number of massive corpuses of electronic resources metadata (journal articles, e-books).

Abes and its networks are committed to produce and share high-quality metadata, to meet end-users needs and to encourage the reuse of this data in other applications. Linking bibliographic data to authority records is a key factor since the creation of the Sudoc in 2000. What was good for the Sudoc is good for other bibliographic databases as well. That is why we made the Abes authority file independent from its original catalogue and consequently open to interactions with other databases. This approach had to be generic by design because we could hardly foresee which partners would be interested in our authority data and services. We believe that open data must be accompanied by open services, read/write services. We call “IdRef” this read/write platform of open data and services dedicated to authority data.

IdRef, heart of a decentralized network around identifiers

When Calames and theses.fr were developed a few years after the Sudoc, the opportunity of a shared authority file came out. It was therefore decided to separate the local authority file from the Sudoc to create IdRef, which became in 2010 the open and independent authority database managed by Abes. IdRef is mainly a generic web interface[3] which enables different client applications from different organizations to search, link to and edit data from their own information system, as if IdRef were a plugin.

If historically libraries have been IdRef's main users, this policy of openness and providing of services associated with a high level of trust brought new players, stimulating the development of interoperability between information systems referring to research publications (institutional repositories) and even research data.

The birth and expansion of IdRef originated from different factors:

- The explosion of scientific publications made critical the quantity versus quality dilemma regarding the description of these documents. It is not feasible to catalog the traditional way millions of articles and chapters released each year, but the very proliferation of author or corporate names makes even more important than before their precise identification with the help of authority files. That’s why authority files must be open, and offer open services, to be usable by other communities and applications, not only traditional library organizations and catalogues. Moreover, to support the exponential growth of agents to identify, the linking process must be automated, without threatening the quality of links. We need reliable algorithms and we need

[1] https://www.istex.fr/
[2] https://www.collexpersee.eu/
[3] www.idref.fr

human interfaces for the data experts to control, correct and complement the machines work. For ten years Abes has been working on this challenge, with IA researchers from GraphIK[4], a research team from Montpellier interested in the entity resolution problem in bibliographic databases[5]. User interface Paprika and quality control program Qualinka are the returns on this long term investment.

- The rise of linked open data stressed the importance of stable identifiers to connect entities coming from different data silos. When Abes began to publish its data on the LOD in 2010, it chose to do it in a distributed way, which conforms to the LOD spirit. Instead of pooling the data of Sudoc, Calames and theses.fr, each catalogue contributed to the web of data separately. But as each of these catalogues is linked to IdRef, they are indirectly interconnected, and connected to other databases (Viaf hence BnF or dnB catalogues, and more). The Abes LOD is a subgraph of the global LOD graph.
- During the same period, other initiatives appeared with the ambition to provide global identifiers for agents involved in scientific or cultural production. Abes is aware that IdRef has a national scope, not a global one. That's why we decided to partner with international initiatives. In 2012, under the historical but now obsolete label "Sudoc", IdRef integrated Viaf together with the Sudoc's bibliographic records. Abes is also a member of ISNI whose identifiers are integrated as much as possible into IdRef records. More recently, through a consortium of 34 French universities or research institutes, Abes became a member of the Orcid Community with the aim of increasing interoperability between the two environments.

Paprika, from the user's perspective

Paprika is a professional user interface dedicated to links quality between the authority database IdRef and bibliographic records in the Sudoc. Its users can create or modify links, i.e. add or modify IdRef identifiers in access points for persons in bibliographic records. It offers an unconventional work environment, adapted to the specific task of reliable identification of entities with the same appellations.

The first version, released in february 2019, is restricted to person entities.

Paprika's users

Paprika first users are catalogers. Among them, we believe that two categories are specifically involved:

- The authority data experts (*"correspondants autorités"*) mandated by each institution (~ 180 people). They are responsible for the deduplication of authority records and the overall quality of the authority file, which also means ensuring the existence and reliability of links from bibliographic records.

[4] GraphIK (Graphs for Inferences on Knowledge) is a joint research team of the French National Institute for Research in Computer Science and Control (Inria, Sophia Antipolis), the University of Montpellier and CNRS https://team.inria.fr/graphik/. Its work focuses on formal representations of knowledge and how to use logical approaches for reasoning based on these representations.

[5] GraphIK and Abes collaborated in two funded projects: SudocAD (2010-2011) and Qualinca (2012-2016).

- Professionals involved in projects dedicated to or including authority linking. Whether the project is limited to the Sudoc environment or whether it is a more open project (for example, identifying in IdRef all researchers from the same institution), Paprika will be a decision support tool complementary to other automated tools.

Paprika is however open to any cataloger in the network with writing rights in Sudoc. We do not want to restrict its access, because we believe it is an open door to discover new tasks and new ways of working with bibliographic and authority data. It is up to each library to decide whether and how to use it, why, by whom.

Main user scenario

The first step is for the user to launch a search with a last name and a first name as inputs. Authority and bibliographic records (with or without links to an authority record) that match are displayed. This search is intentionally broad and may retrieve results that seem not relevant because noise is preferred to silence: it is important to be sure that all relevant authority records are retrieved to avoid the creation of a duplicate record. Fortunately the user can filter out the irrelevant results by unselecting some names. If he does so, all the *reference in context* (RC) and *reference as authority* (RA) having this appellation as name and surname are hidden. We call “initial partition” this subset of RAs and RCs resulting from the initial search and the possible filtering choices of the user. Initial partition includes links between RC and RA, although RC may not have any link to and RA.

It is important to stress that what is displayed is not really bibliographic records but references to a person *as a contributor (author, etc.) of a document*. We name it *reference in context* (RC), this context being the bibliographic description of this document. It is close to the classical notion of “access point. Chein, Gutierrez and Leclère[6] proposes a definition for this “contextual description of a person” : “a set of metadata extracted from the description of an object in which the entity plays a role”. An RC differs from what we call *reference as authority* (RA), or a “referential description of a person” according to Chein, Gutierrez and Leclère[7], which is “a reference to a person embodied by an authority record in a referential database or authority file”. The RC perspective, absent from standard cataloging tools as well as online library catalogs, is highly needed to control the validity of bibliographic links. Actually, the main user task is to compare RC with RA in order to decide whether the RC and the RA refers to the same person. According to this judgement, the user validates or corrects existing links and creates links when missing.

We invite the user to forget for a while the look and feel of his usual cataloging environment, which has not been made for data curation, but for manual unitary cataloging. This means leaving behind the traditional bibliographic record. In Paprika, RCs and RAs are materialized by boxes: a large box represents an RA while a small box represents an RC (see figure 1 and figure 2). This graphic choice is convenient to display a large number of RAs and RCs on a unique screen because boxes are easily resizable. Moreover, there is enough space to see

[6] Chein, Michel, Alain Gutierrez, and Michel Leclère. “A General Framework to Build and Assess the Quality of Authority Links.” In *Proceedings of the 10th International Conference on Knowledge Capture (K-CAP ’19)*. 13-20. New-York: Association for Computing Machinery, 2019.
[7] Chein, Gutierrez, Leclère 2019

first-level information: document citation and role for RCs, appellation and sources for RAs, but also information recalling the actions performed by the user. Boxes also suggest to the user that he can get more information by opening them: co-contributors, publication date, subject headings or publisher (see figure 3). For a given RA, this content is built out of all the information relating to the RCs which are linked to it.

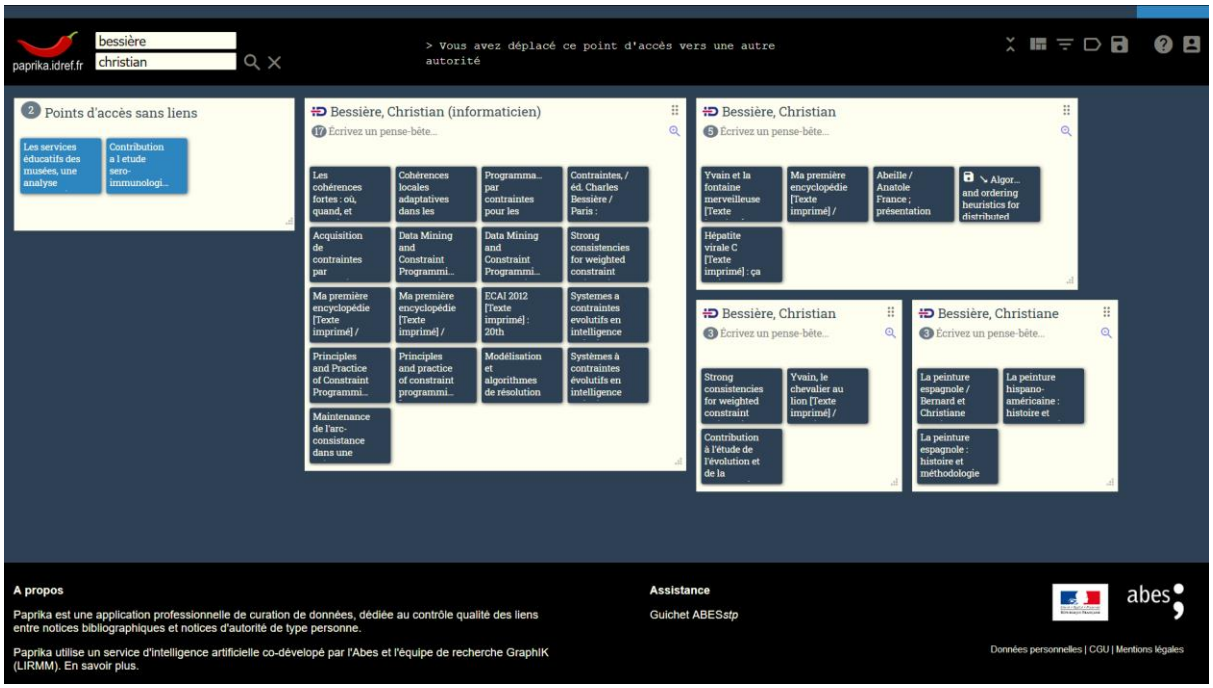


fig. 1: Result displayed after the initial search for "Bessière, Christian".

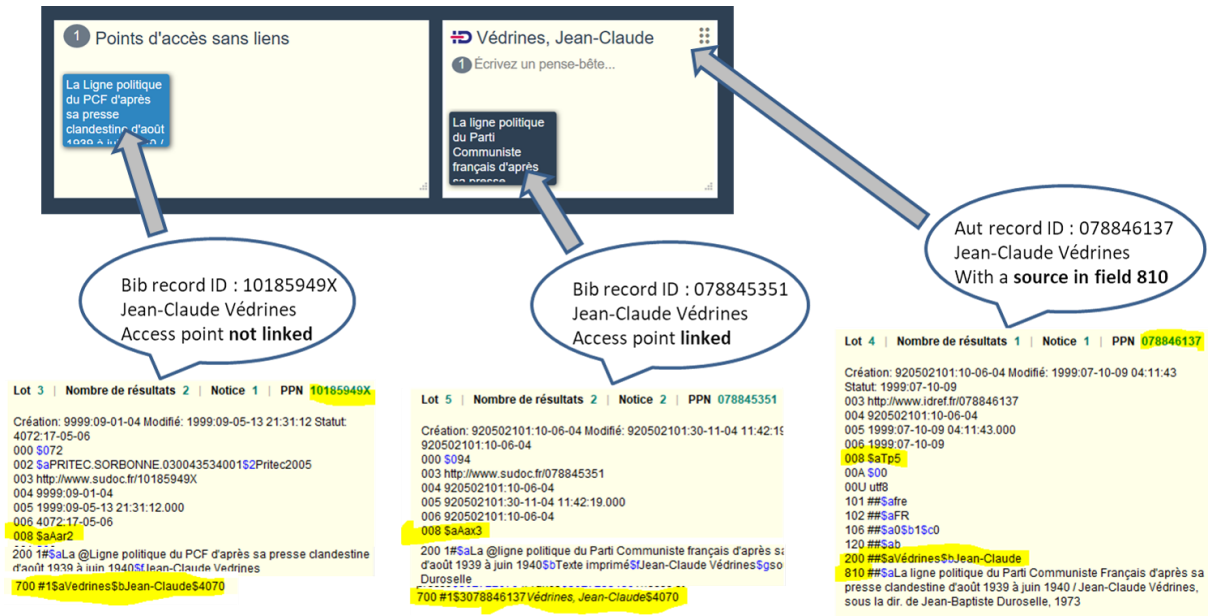


fig. 2: Illustration of the notions of RC and RA, in comparison with traditional bibliographic and authority records (in UNIMARC format).

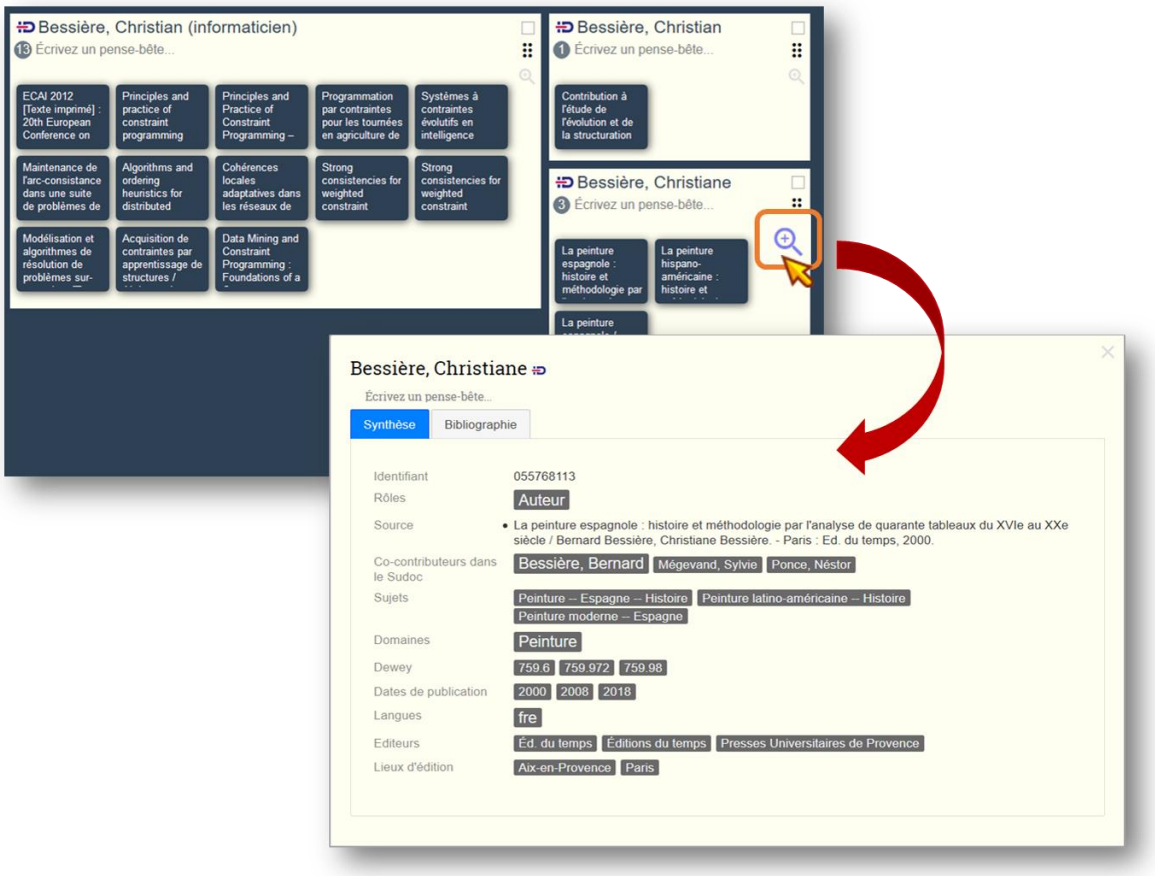


Fig. 3: Detailed visualization of an RA

To create or modify a link, the user simply moves, by drag and drop, the examined RC into the RA of his choice. This does not by itself update the Sudoc: saving the modification is an additional specific and explicit action.

Paprika uses the IdRef iframe, which allows the main functionalities of IdRef to be integrated into any web application. If no RA exists for a given RC, the user can create a new RA box, from which he will be redirected to the Idref creation form to create a new authority record. Once done, the new box is automatically filled with information from the new record.

We demonstrated that Paprika offers a comprehensive work environment that encompasses all the tasks related to the quality control of links, creation of new authority included. It is supposed to be a self-sufficient work environment.

Towards an UX design approach

As Paprika features and objective are quite distinctive, we felt free to imagine an application that would depart from traditional bibliographic interfaces appearance. Our long term involvement in IdRef activities and community helped us to define guidelines for a decision support tool dedicated to entity resolution problem in bibliographic databases. In addition, we collaborated with experts in authority data, taking into account the shortcomings of current professional tools and the resulting practical limitations. One of the main objectives was to create an easy-to-use tool thanks to which users could follow a complete path on a single

screen. The application must meet the usability criteria: not only must it allow the user to carry out his tasks (effectiveness), but it must also facilitate his work, save time and efforts (efficiency), and offer him a comfort of use, even fun (satisfaction). To reach this objectives, the development team used different methods to get feedback from users.

Usability testing

We adopted a light usability testing method proposed by Steve Krug[8]. It is a pragmatic approach to find and fix usability problems even with a low budget. The method consists of asking 3 users to test the interface for one hour each following a predefined scenario (a list of tasks). According to Steve Krug, “for “the do-it-yourselfer” three is sufficient to fix as many problem as possible”. He follows and adapts the recommendations of Jakob Nielsen[9] who showed that “Elaborate usability tests are a waste of resources. The best results come from testing no more than 5 users and running as many small tests as you can afford.” A guide helps the user to complete the tasks while thinking aloud. In another room, the project team watches the user's screen and hears his comments and feelings about the interface. At the end, the team discusses the different problems and makes a concrete plan to solve the most important ones. We choose it because it was easy to settle: it only requires half a day for the team per session and limited material. The advantage is also that every stakeholder of the project (developer, product owner, manager) is proactive, as action plans are decided together.

So far, three sessions took place at three key steps of the development of the first version. The testers’ panel was composed of data experts from Abes, authority data experts and ordinary catalogers. Each time, usability problems were detected. It allowed the team to fix the major ones by quickly adjusting the interface or by redesigning it more deeply. Moreover, users also claimed for new functionalities, which were added to the product backlog for future versions. In conclusion, those tests were extremely beneficial to validate the main functionalities, perceive where the user’s path was broken and why and set an action plan to fix problems within the limits of our means.

Qualitative Interviews

Paprika is a brand new interface that is not yet integrated with the general working environment of librarians. To find out how Paprika is perceived and how it integrates (or not) work habits and why, we have started to interview different librarians, who are using Paprika more or less regularly. We hope to define profiles or personas based on various needs, experiences, behaviours and goals.

Qualinka, the “Paprika effect”

Qualinka feature is a quality control program which processes the attributes of RCs and RAs (surname, first name, life dates, sources, subjects, co-contributors, publication dates, roles, etc.) to diagnose the quality of existing links. Once Qualinka provides results, the diagnosis is

[8] Krug, Steve. Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems. San Francisco: New Rider, 2009.
[9] Nielsen, Jakob. «Applying discount usability engineering.» IEEE software 12, 1 (1995): 98-100

visible on the screen as colours applied to the RCs (linked or not linked) and connectors between RCs (linked or not) and RAs (see figure 5). A green RC box shows a correct link, a red one shows an incorrect link and an orange one shows a doubtful or unknown link. A connector going from a red RC to an RA distinct from the initial one shows the RA deemed to be the right one for this RC. In the context of Paprika, Qualinka's diagnosis is just a helper and doesn't make automatic changes. The user has still to move RC boxes to create or correct links.

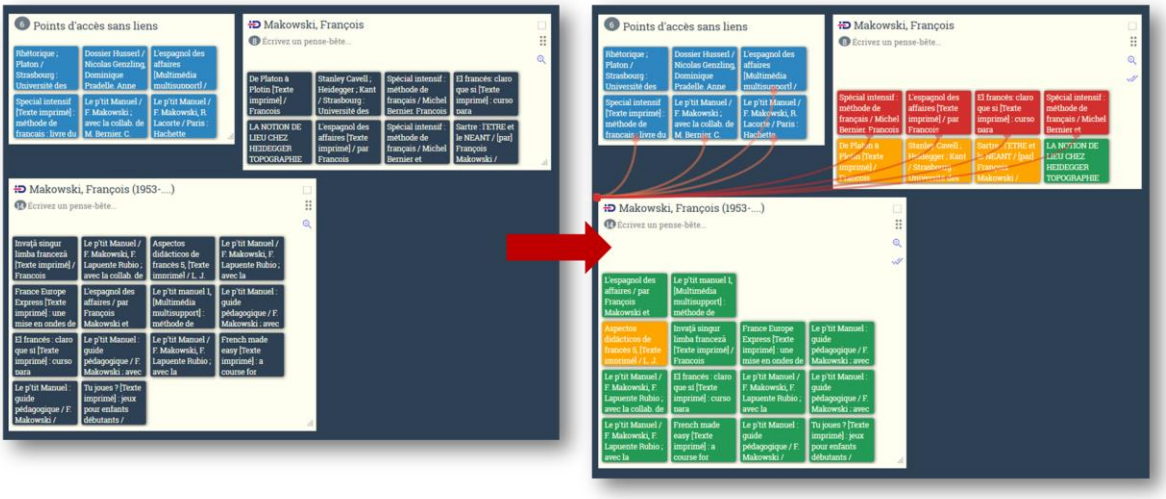
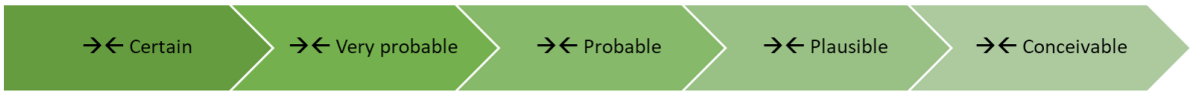


Fig. 4: Left - the initial partition / Right - the partition as diagnosed by Qualinka

Note that Qualinka and Paprika are loosely coupled: the interface could call another quality control program. Reciprocally Qualinca can be used outside Paprika, as a web service or in a batch mode.

Expert knowledge representation

Co-reference rules : 5 trust values



Example (X = RC, Y = RA)

If X's *form of name* is homonymous to Y's
and if the *corporate body* associated with X is identical to one of those associated with Y
and if the document associated with X has *keywords* which are very close to those associated with Y,
THEN is is **Very probable** that X and Y are one and the same person.

Difference rules: 4 trust values

Example (X = RC, Y = RA)

If X's *form of name* is only close (and not homonymous) to Y's,
And if the document associated with X has different Dewey indexes from those associated with Y,
THEN is is **Very probable** that X and Y are not the same person.

Fig. 5: Qualinka process

Qualinka process can be described as follows (see figure 6). The starting point is an initial partition composed of RCs and RAs, composed of links the user wants to evaluate **(A)**. During the development of Qualinka, tacit knowledge and reasoning of librarians **(B)** has been formalized as logical rules expressed in the declarative logic programming language *datalog* and through criterions and attributes[10]. An attribute is an element of knowledge associated with a reference (RC or RA), while a criterion is a function that compares the values of certain attributes to obtain a proximity or distance score between an RC and an RA. A rule declares an assertion on the relation between two references based on a set of conditions embodied by the criteria (see figure 7). A rule can be used to conclude that there is closeness (co-reference) or distance (difference), with varying levels of intensity **(C, D)**. The initial partition is analyzed by Qualinka, i. e. every RC is compared to every RA and submitted to the rule engine. When a rule is true for an RC-RA couple, a relation is created between them **(E)**. In the end, based on the nature (→← co-reference or ←→ difference) and degree of trust of the relations, links are computed as safe links, suggested links or impossible links **(F)**.

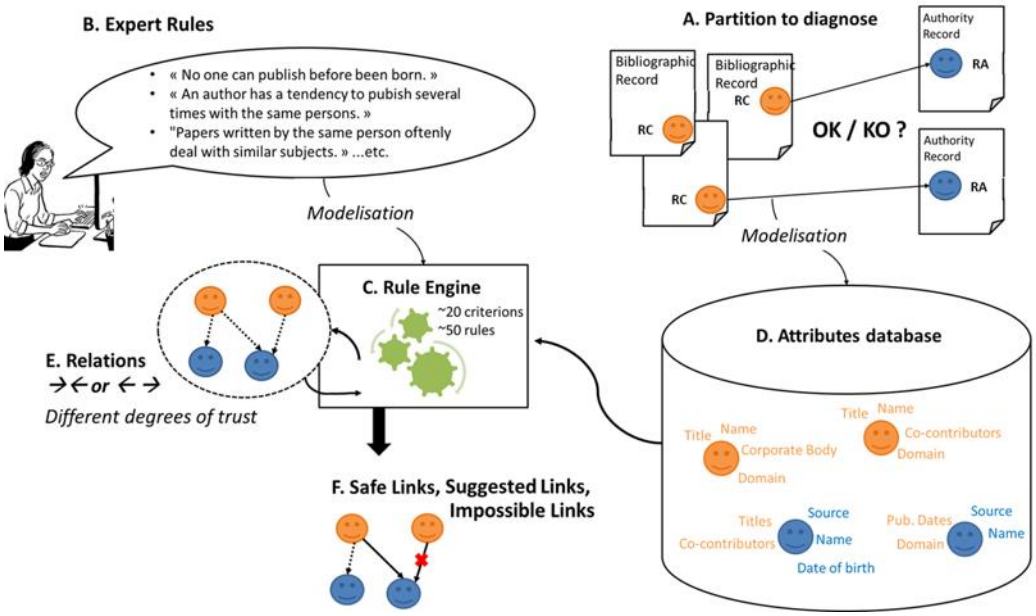


Fig. 6 : The trust values for co-reference / difference rules

Qualinka carries two key concepts:

1. The concept of "super-authorities" : The attributes of the RC linked to an RA are viewed as attributes of this RA
2. The iterative computation of links in *n* steps. At first step, Qualinka discards all existing links except some rare links that can be considered as safe. A link between an RC and an RA is considered as a safe link only if the RC comes from a document mentioned as a source of the RA or if the RC and the RA have the same creation date. These first safe links are essential as they allow to initialize the enrichment of RAs with RC's attributes, so that new links can be computed in the next step. Each new step potentially produces new safe links, which increases knowledge about RAs by adding more attributes from RCs. In this way, Qualinka avoids calculating links from incorrect

[10] See Chein, Gutierrez, Leclère 2019 for more details.

information obtained from existing false links. This prudential iterative method is a key factor to get high quality results.

This combination not only creates reliable links for RCs that have no links, but also offers a diagnosis on existing links. The latter is provided in the form of a status declared for every RC of the initial partition[11].

Qualinka fits into the category of symbolic artificial intelligence (AI) tools. Considering our initial problem, this “Good Old Fashioned AI” (GOFAI)[12], in comparison with connectionist AI or machine learning[13] is still interesting because it makes the results understandable by human experts (explainable AI).

It was evaluated[14] that in over 70% of cases SudoQual made the same choices as a human expert with over half being certain choices (safe links) and only made mistakes in 0.3% of cases. For the rest (less than 30%), SudoQual could not give a diagnosis, either because of a lack of information associated with RC and RA or because no authority corresponded.

Beyond Paprika, the framework for wider use

Actually, Qualinka is a particular application of the standalone framework SudoQual. SudoQual has been developed out of a prototype Abes and GraphIK team research (LIRMM, Montpellier) built during the ANR funded project Qualinca (2012-2016). In 2018-2019, Abes partially rewrote this prototype to get a rich framework dedicated to linking tasks capable of handling a variety of use cases.

Paprika is one and the first use case that integrates Qualinka, but Abes and Abes users face more needs that go beyond quality control in Sudoc and IdRef:

- Automatic linking between RCs with no link and RAs
- Clustering of RCs
- Alignment between RAs (IdRef and ORCID, by instance)
- Duplicates detection

These tasks can potentially be applied to any bibliographic databases (RC databases) and authority files (RA databases). They can be integrated into a user interface or called as standalone web services or batch commands. At last, they can apply to more types of entity than persons.

A decision support tool

Paprika and Qualinka are complementary. The user can decide to ignore the Qualinka functionality, but he can also take it as a mere helper, not an oracle. Qualinka results can be viewed as suggestions, which can be bypassed or accepted. Or the user can trust Qualinka enough to let it do the easy and boring part, and focus on the hard cases.

To be trusted by an expert, Qualinka results have to be explainable, not only accurate. It is the case because the algorithm relies on explicit attributes, criteria and rules, not on a

[11] Chein, Gutierrez, Leclère 2019.

[12] Haugeland, John. Artificial Intelligence : The Very Idea. Cambridge: MIT Press, 1985.

[13] Minsky, Marvin L. “Logical versus analogical or symbolic versus connectionist or neat versus scruffy.” AI magazine 12, 2 (1991): 34-51.

[14] Chein, Gutierrez, Leclère 2019.

statistical black box. The data sent by Qualinka contain its conclusions but also the “reasons” why it concludes this way. We are considering showing these “reasons” to the user, but we have to think carefully about the way to do it. Too much explanation can yield a cognitive overload or paradoxically arouse skepticism about the reasoning of the machine and then about its results, even when they look intuitively correct. We could just lay emphasis on the attributes that have been decisive in the reasoning.
Qualinka helps the user but this partnership can be symmetrical. Indeed, Qualinka can remain mute if information is lacking and the iterative calculation cannot initialize. In this situation, all RC boxes will be orange. Instead of examining each RC and RA, the user can focus on one or two RC, validate or invalidate one or two links and call Qualinka anew. These links will be considered as safe initial links and the calculation will get enough input to launch the iterative process. It is a kind of snowball Paprika effect.

Paprika, from the developer’s perspective

Paprika was designed and developed as a single page application. The interactions between this client and the server(s) occur only through web services. The function of these web services is to send or retrieve data.
This implementation design helps to make Paprika a generic application: the current web services called by Paprika could be replaced by other services. The integration of these services is totally independent from the server where the HTML, JS and CSS pages are hosted. The web services called have just to be CORS (Cross-Origin Resource Sharing) enabled so that any web client can call them from a different domain.
Paprika currently integrates 9 different web services, all run by Abes, but one.

1. A web service to get the “initial partition”, i.e. the list of RCs and RAs that match the initial search, and the links between RC and RA. A first name and a last name are the search parameters. The initial partition is used to build the main structure of the page: RC boxes inside of RA boxes, and a box containing RCs with no link. We don’t know anything about these RCs and RAs, except their identifier and their links. That is why these boxes are empty, as long as RC and RA attributes have not been obtained.
2. A web service to get the attributes of each RC (title, co-author, date, etc.). This service (and the following one) is asynchronous, so that the user can browse the page before the retrieval of all attributes.
3. A web service to get the attributes of each RA (names, dates, sources, etc.).
4. A web service to extract the “topics” out of the titles of RCs linked to an RA. These new RA metadata are generated by https://www.textrazor.com, that offers text mining as a service. These topics are not integrated in the RA attributes. They are just displayed in the box. The user can consider these keywords as a kind of clue, or summary of the RA, but she can also erase them, or replace them by her own keywords.
5. A web service to send the initial partition to Qualinka.
6. A web service to retrieve the diagnosis computed by Qualinka. This diagnosis updates the RC boxes (colours, icons).
7. A web service to sign in using the Sudoc login.
8. A web service to save the new or updated link in the production database of Sudoc (write service).
9. A web service to save the actions of the user in a log (write service). Data saved in this log can be used for different purposes: roll back ; study of user actions ; identification

of more reliable links between bibliographic record and authority records, these links having been asserted or confirmed in the quality control context of Paprika, not during daily cataloguing.

For Paprika to be truly generic, it should be able to integrate new sources and new targets without any change in the main source code. Each new source or target would be plugged into Paprika, through a kind of advanced configuration file. This configuration file would specify the web services to call, the paths or query to extract the relevant attributes from the results and instructions for the display in the interface. Today, a kind of attributes model is built in Paprika, empirically specified by the availability of metadata in Sudoc and IdRef. But Paprika should not demand any specific model of metadata. It should be up to the provider of a new source or target to specify which attributes are attached to RC and RA and how they should be stored (string or array) and displayed (list or tag cloud, by instance; which label). It would be much more complex but still possible to add to this configuration file the instructions that would make possible the application of Qualinka to this source/target pair.

It seems to be quite straightforward to *query* any bibliographic source through its API, but it is less probable that any source would allow the update of the links stored in its database from Paprika. Some will, provided that only authorized users can update the remote database. In the absence of the remote update option, one can imagine an alternative way to save the results of the user's actions: a detailed report would be generated by the interface and saved by the user. This report would be structured and then processable. As a last resort, the public RDF database data.idref.fr[15] could store and publish itself the links to IdRef that have been corrected or created in Paprika, but the sound workflow is for the bibliographic source to integrate the new or corrected links and for data.idref.fr to synchronize with this source.

The SPARQL standard could be the best technical candidate to implement read/write interactions between Paprika and any bibliographic source in a generic way. As the protocol and the query language would be common to all compliant databases, the Paprika configuration file would just contain the endpoint url and the queries. Indeed, the information about how to store and display such and such attribute in Paprika could be conveyed in the result of the query itself, without need of further specification. Alas, this promising generic solution has two serious drawbacks :

1. Full text search is not supported by the official SPARQL standard, although specific implementations exist.
2. Bibliographic RDF databases are generally mere copies of the production database. Updating the copy has no sense. Moreover this copy is too often not synchronized with the production database, which is a problem for the search step, not only the update step.

The main challenges were design issues, more than technical issues (fortunately because the developer is a librarian). Paprika is a quite specific professional interface designed to perform complex tasks on a set of data units that depart from the traditional packaging of bibliographic or authority data as records. The user must be able to zoom in and zoom out, to switch

[15] data.idref.fr (https://data.idref.fr/) is the freely accessible IdRef triple store. It offers search functionalities synchronized with Sudoc and IdRef production environments, making it possible to obtain bibliographic and authority data and their relationships in RDF.

between an overview of the many boxes and a focus on individual boxes and their attributes. The initial search may retrieve dozens of RA boxes and hundreds of RC boxes.

We describe below the design solutions adopted:

- To get an overview or to focus, the user has just to use the browser zoom, not an *ad hoc* zooming functionality.
- The relative positioning of boxes is basic in order to be predictable: each box is floating at the right of the precedent one, unless it has to be pushed to the next line. There is no magical optimization of the positioning to render a pretty mosaic aspect, where the space between the boxes would be minimized. The box should not wander in an unpredictable way, which would prevent the user to find the box she is interested in. She is able to drag and drop the RA boxes according to her preferences (e.g. to put closer RA boxes that look similar). She keeps control.
- The RA boxes can be resized, one at a time or all together.
- As the initial search is by purpose broad and fuzzy, it often retrieves some noise. The user can decide to filter out some RA (and RC with no link) by checking out their appellation.

Conclusion

Paprika and Qualinka are pieces of a toolbox that Abes designed and developed to achieve its ambitions related to the repurposing of our traditional authority file as a national provider of reliable identifiers for the French Higher Education & Research community. This community is composed of numerous organizations and data silos. IdRef efficiently increases the interoperability of these data silos when it is used to identify the scholars across bibliographic catalogues, open or institutional archives, bibliometric applications, digital libraries, courses catalogues or human resources information systems. To achieve this goal, IdRef offers a comprehensive set of open data and open services. Some of these services are read only services: search interface and API, lookup web services, synchronization web services to list recently created, modified or merged records, etc. The fundamental and distinctive initial choice has been to encourage partners not only to reuse IdRef data but also to update the database. IdRef is constitutively a collaborative identifiers database, enriched by a diverse network of metadata experts.

As it does for its other networks (Sudoc, Calames, theses), Abes works to make easier and more efficient the production and control of authority data, through human support, web services and web applications. We strive to reach an optimal equilibrium between of quantity and quality. Qualinka was designed to diagnose the existing links but also to automatically generate links to IdRef identifiers when links do not preexist, as it is the case in publishers data massively imported in Sudoc or other databases. Paprika was designed for the data expert to check and enhance the quality of existing links, with or without Qualinka assistance. Both are complementary to one another and to the other pieces of the IdRef toolbox.

Paprika and Qualinka were designed and developed to serve the needs of the Sudoc and IdRef professional users, but we want it to be generic. Instead of or in addition to Sudoc, more bibliographic sources could be integrated in the interface. Let’s mention the digital library Persée who is already densely linked to IdRef, but also other bibliographic sources that are

sparsely or not at all linked to IdRef (institutional repositories). Instead of IdRef, other authority files (or more generally person reference databases) could be integrated in the interface and displayed as RAs. Let's mention ISNI or ORCID. It is also true of other types of RAs, as corporate bodies, works, etc.

Beyond library data, Paprika and Qualinka could be employed to perform analogous tasks in other domains. For instance, in the context of named entity extraction in text mining, an agent could use Paprika to check if the program has rightly identified the type or the identity of a term. Even farther from libraries or digital humanities, we know of a project from the biomedical domain that plans to use the generic framework to link raw biological observations to LOINC (Logical Observation Identifiers Names and Codes), "the international standard for identifying health measurements, observations, and documents".

Actually, Paprika and Qualinka are potentially useful tools in any domain where the classical "record linkage" task applies. We hope that eventually a community (or two) will emerge around Paprika and Qualinka, which we plan to release as open source softwares in 2020.