OPEN ARCHIVES INITIATIVE: A FAST WAY OF INTEGRATION INTO GLOBAL OPEN SCIENCE

Yurii Tkachov
ORCID: https://orcid.org/0000-0003-1556-2463
Oles Honchar Dnipro National University, Dnipro, Ukraine

INTRODUCTION

The idea of creating a technology for collecting metadata from open archives dates to the late 1990s. At that time the scientific community was faced with the issue of disparity and inaccessibility of scientific data, which made it difficult to exchange knowledge and collaborate between researchers. To solve this problem **Open Archives Initiative** (OAI) was launched, which aimed to create an open, standard and interoperable platform for the collection and exchange of metadata (*Open Archives Initiative*, n. d.).

In 1999, the first meeting of representatives of libraries, archives and publishers was held in Santa Fe, New Mexico, where the basic principles and requirements for the new technology were agreed upon. The main goal was to create a protocol that would allow different archives to easily exchange metadata without the need to create complex interfaces or adapt to the specific systems of each archive.

The result of this meeting was the development of **Open Archives Initiative Protocol for Metadata Harvesting** (OAI-PMH), which was officially published in 2001. The protocol provides a simple and standardized mechanism for harvesting metadata from different archives and providing access to them through a single interface.

The main directions of implementation of OAI-PMH were as follows:

– ensuring open access to scientific data and the possibility of free exchange of metadata between archives, which would facilitate the availability of this data for all interested parties;

– ensuring data interoperability by developing a single standard for metadata that would allow easy integration of data from different sources;

– improving scientific communication through knowledge sharing and collaboration between researchers by providing access to relevant scientific publications and data;

– stimulating the development of tools that meet OAI specifications and help developers create new services and tools that work with various data sources and contribute to the development of open science.

Since its inception, OAI-PMH has undergone several changes and improvements aimed at increasing its efficiency and adapting to new technological requirements. The protocol has become the basis for many modern open science platforms, providing integration and access to scientific data worldwide.

However, like any technology, OAI-PMH has its challenges. The protocol implementation process requires specific technical knowledge and resources, as well as adequate support from organizations. Despite this, OAI-PMH remains an important tool for the development of open science and scientific communication in the modern world.

Undoubtedly, the scientific community has a need for open access to scientific data and publications, which makes the implementation of the Open Archives Initiative Protocol for Metadata Harvesting an important tool for supporting open journals, conferences and repositories. This protocol allows you to efficiently collect metadata from different sources and guarantee their compatibility. However, the process of implementing and integrating OAI-PMH is not without problems, many organizations face various challenges along the way.

PURPOSE

This study aims to analyze and summarize the practical experience of deploying and integrating platforms for open journals, conferences and repositories with support for Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), highlighting errors and difficulties that may be encountered by managers and administrators of journals and repositories, as well as conference organizers, and offering recommendations for preventing and avoiding these problems.

RESEARCH METHODS

The research methodology includes the analysis of real cases. Applied methods of testing, validation of the OAI-PMH protocol, observation, study of regulatory and technical documents.

One of the key aspects of the research was the study of the most common free software for open repositories and journals. This software plays an important role in providing support for the OAI-PMH protocol and facilitates the availability of scientific data.

OVERVIEW OF PLATFORMS SUPPORTED BY OAI-PMH

Determining the prevalence of software for the deployment of open institutional or scientific repositories turned out to be a very easy task (fig. 1) according to the distribution rating of the OpenDOAR catalog (*OpenDOAR Statistics - Sherpa Services,* n. d.).

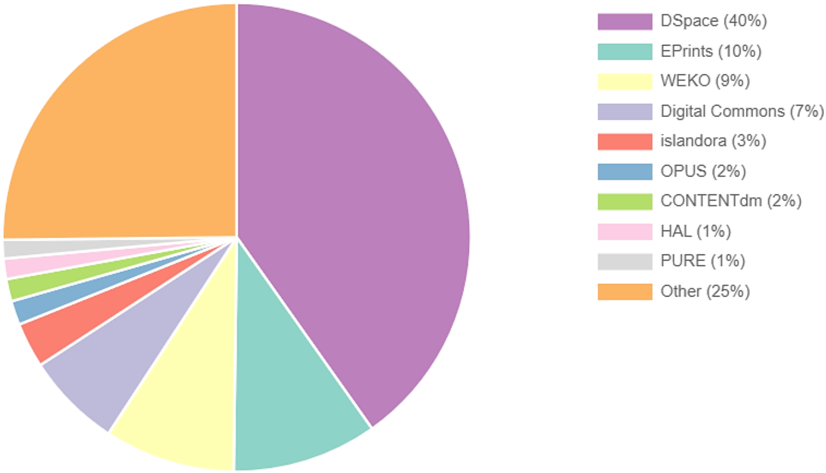

Figure 1 – Software platforms overview according to the OpenDOAR

Therefore, the undisputed leaders in the segment of repositories, which collectively cover 50% of use cases, are DSpace and EPrints.

DSpace is an open-source platform for creating and managing digital repositories, developed at the Massachusetts Institute of Technology (MIT) and widely used by libraries, universities, academic institutions, and other organizations to store and provide access to their digital collections. DSpace is suitable for storing a wide range of digital resources such as articles, books, images, audio and video recordings, datasets and much more. This system offers a wide range of functions that facilitate the collection, cataloguing, preservation and provision of access to digital collections (*DSpace Home - DSpace,* n. d.). DSpace is used by institutions around the world, including the Massachusetts Institute of Technology, Harvard University, the National Library of Australia, and the Library of Congress.

The main characteristics of DSpace can be noted:

– it's open-source software, which means it's free to use and modify, making it an affordable option for organizations with limited budgets;

– DSpace can be customized to meet the specific needs of an organization, the software offers a wide range of features that can be turned on or off, and can be extended with plugins and modules;

– such a repository can scale to support any size digital collection, from a small number of items to millions;

– it is a reliable and sustainable platform capable of storing and providing access to digital collections;

– this platform supports a wide range of standards, including Dublin Core, MARC21 and OAI-PMH, making it an indispensable tool for integrating with other systems and sharing digital collections.

EPrints was one of the first platforms for open-source repositories. Many universities and research institutions started deploying it in the early 2000s, so it has a long history and a large user base, developed at the University of Southampton (UK) and widely used by libraries, universities, research institutions and other organizations for storage and access to their digital collections. EPrints is a powerful and flexible platform that can be deployed to store a wide range of digital assets, including articles, books, images, audio and video recordings, datasets and more. It offers a wide range of functions that facilitate the collection, cataloguing, preservation and provision of access to digital collections (*EPrints Services*, n. d.). EPrints is implemented by organizations around the world, such as the University of Southampton (UK), University of Manchester (UK), University of Queensland (Australia), University of Toronto (Canada).

I can also focus on covering the main features of EPrints after the features of DSpace, but you're guaranteed to have a sense of déjà vu afterwards. It's only logical because DSpace and EPrints really do provide the same management capabilities for digital repository collections.

What is the phenomenon of the popularity of DSpace and EPrints, and at the same time, why is there such a significant discrepancy, if you look at the statistics of implementation (fig. 1), 40% and 10% is a noticeable difference, and what exactly are the differences?

1. Number of supported language localizations.

According to official sources, both platforms DSpace (*Features - DSpace*, n. d.) and EPrints (*Browse by Type - EPrints Files*, n. d.) support 22 interface languages, but this is not an exact number, because some unannounced language localizations are in the process of development, they are available for download, but may be partially completed. In conclusion it can be argued that both platforms are the same in this respect. The number of language localizations really matters and can significantly influence the choice of software for open repositories. Native language support makes the system more attractive and convenient for users, promotes its spread and successful implementation in different countries.

2. Attractiveness of frontend visualization.

In fact, repositories differ from file archives, digital libraries, other resources for hosting various content, because they are purely technological resources, the main function of which is to ensure the correct distribution of metadata according to the OAI-PMH protocol to display them correctly in scientific search engines such as Google Scholar, Crossref, OpenAIRE, DOAJ, BASE and others.

Therefore, the presence of advanced search functions, attractive and visual presentation of data, filtering and structuring of resources according to certain characteristics are secondary. In other words, repositories may have limited functionality in the context of data presentation, compared to digital libraries or file archives, which are focused on providing access to collections of digital materials such as books, articles, images, but do not necessarily support OAI-PMH.

However, the external interface (frontend) of any information system should be oriented towards people, not machines, that is, it should have the usual quality indicators, at least aesthetics and ergonomics. In this context, EPrints is significantly inferior to DSpace. EPrints frontend looks dated and unattractive to say the least, unlike DSpace which looks modern and well adapted for mobile viewing.

3. Intensity of project development.

Project development and regular updates that add new features and improve user interaction are one of the factors that influence the popularity of the software. If you look at the history of EPrints versions (*History - EPrints Documentation*, n. d.), during the years 2011–2018 the project did not develop, and from 2018 to 2023 only small incremental updates of the current version were released. Consequently, the lack of active development and regular updates of EPrints for many years has negatively affected its popularity, especially in terms of competition with other, active projects.

If you look at the development of DSpace (*Releases - DSpace - LYRASIS Wiki*, n. d.), you can see that this software is very actively developing, trace the development prospects and even plans for future releases.

4. Quality of technical documentation

Almost any community-driven open-source project is characterized by poor documentation. In this context, DSpace (*Installing DSpace - DSpace 7. x Documentation - LYRASIS Wiki*, n. d.) and EPrints (*Category: Installation - EPrints Documentation*, n. d.) are not exclusive. To begin the deployment of any web platform, the system administrator needs to understand at least, first, the server software requirements, which are generally absent in the documentation, and second, it is desirable to have a rough step-by-step guide for installing and deploying the system. Unfortunately, in both cases, this part of the documentation is written very poorly, there is no clear logic and sequence.

The only alternative is to find articles on thematic sites or blogs that are written by those who have already covered this path - the community of users and developers often becomes the main source of support and documentation for open-source projects. They create forums, blogs, video tutorials and other resources to help new users learn such systems. The lack of official documentation can be partially compensated by the resources listed, but it also creates unequal access to knowledge, as not all users can find or understand unofficial materials. Clear and logically structured step-by-step instructions help reduce the time and effort required to deploy the platform. The absence of such instructions forces system administrators to spend more time searching for information and solving problems, which can reduce the effectiveness of software implementation.

5. Software implementation.

Anyone who works in the field of network technologies understands that LAMP is the de facto industry standard for deploying a web server, it is a software bundle, also called a stack. LAMP is an acronym for Linux, Apache, MySQL, PHP (*Что такое стек LAMP? – Объяснение стека LAMP – AWS*, n. d.). Alternatives to PHP in some cases may be Perl or Python. It is necessary to add to this information some technical details regarding the link that ensures the execution of external backend applications on the server side, their interaction with the database server and the visualization of the frontend - the graphical user interface in the form of dynamic website pages during HTTP / HTTPS requests, as well as compare them It is common knowledge that all programming languages are either compiled or interpreted (freeCodeCamp.org, 2020), sometimes they may partially combine both technologies. Comparing three programming languages (Java, Perl and PHP) in relation to the studied platforms, it can be stated that:

– Java is compiled into machine code before execution, making it the fastest of the three languages, may be the best choice for developing large and complex enterprise-level systems, as the language offers high scalability, reliability and security;

– Perl has a compiler-like interpreter that generates virtual machine code before execution, making this language slightly faster than PHP, it may be a better choice for tasks that require regular processing of textual data as it has advanced capabilities;

– PHP interprets line by line, making it the slowest of the three languages, but it can be a better choice for developing websites and web applications because it has a wide ecosystem of web development frameworks and tools.

Returning to the study of DSpace and EPrints, the following can be argued. DSpace is an enterprise-level platform that dynamically develops and updates. Thanks to the use of Java, the programming language on which this system is built, DSpace provides reliable and fastest processing of collections consisting of thousands, tens, even hundreds of thousands of resources, which really makes it popular.

However, not all institutions can afford to deploy and operate such a system with limited resources for the following reasons:

– a non-standard, arguably uncommon, and even unique, server software stack;

– to ensure reliable and productive functioning of the system, ideally, at least three servers should be put into operation - a server for the backend of the system, a server for the frontend of the system, and a server with a reverse web proxy that provides external interaction; whether it will be physically one computer or several, it depends on the institution's budget and the projected amount of data that will be stored in the future.

EPrints is a platform that provides similar functionality when compared to DSpace, but practically without the above-mentioned disadvantages. Installation requires an almost standard LAMP server software stack, except Perl instead of PHP. Moreover, Perl turns out to be more productive than PHP, and this is undoubtedly a positive factor. In addition, such a web server with adequate performance can coexist with other systems that require a LAMP stack, for example, WordPress for an institution's website or a news blog (*Download,* n. d.). This option is economical, because for certain technical reasons it is possible to avoid the mandatory commissioning of a separate server.

In this study, I will not focus on a detailed analysis of other well-known platforms for open repositories that occupy another 25% of the distribution, but I will certainly explain why. Common reasons are proprietary, commercial, lack of language localizations or limited support for these systems. However, I will focus on one more system, which is not a platform for open repositories, but its principles of operation and the main function of providing open access to scientific data according to the OAI-PMH protocol are similar – this is **Open Journal Systems (OJS)** (*Public Knowledge Project*, n. d.).

The main difference between platforms for open repositories and platforms for open journals is precisely the presence or absence of a system that ensures the process of submission of scientific papers and the workflow of publishing.

The main function of repositories is to store and provide access to scientific materials such as articles, dissertations, reports, research data, etc. They provide long-term archiving and easy access to these materials. Repositories do not have integrated tools for managing the process of submitting articles, their review and editorial work. This means that authors can upload their work directly to the repository, and these works usually do not go through a formalized review process before publication.

Platforms of open journals have comprehensive tools for managing the entire workflow of publishing activities, which consists of the following stages:

– submission of articles through the system;

– reviewing, in particular with the possibility of appointing anonymous reviewers to evaluate and provide feedback on submitted works;

– editorial work, which ensures the process of reviewing, accepting or rejecting articles;

– communication with authors at all stages of the publishing process;

– publication of the article on the journal website after review and editorial processing of the article.

I was unable to find data that would clearly demonstrate quantitative indicators of the use of software platforms for open journals in the same way as for repositories (fig. 1). However, a 2021 study by the OA Diamond Journals team, shows that about 60% of open access journals work on Open Journal Systems. OJS covers more than 34,000 journals worldwide, in about 150 countries, with more than 6 million articles in more than 60 languages. This platform is also selected for several national open access publishing portals (*Solution: Open Journal Systems (OJS) | Infra Finder*, n. d.).

The key to the success of the Public Knowledge Project (PKP) for the past two decades has been its commitment to writing, maintaining and releasing free and open-source software for publishing platforms and workflows – namely Open Journal Systems (OJS), Open Monograph Press (OMP) and Open Preprint Systems (OPS) which support access to the most modern open scientific publications and fully comply with OAI-PMH specifications. The following main characteristics of OJS can be outlined.

1. Number of supported language localizations.

According to official sources, OJS fully or partially supports 113 interface languages at the time of writing. The translation process is in constant development and varies in degree of completion (*Open Journal Systems*, n.d.).

2. Attractiveness of frontend visualization.

OJS currently supports 6 user interface themes that are available to download, install and update through the plugin gallery, and there are other themes that can be downloaded from third-party resources. Each of the themes offered has a greater or lesser number of settings. In addition, the possibility of more flexible debugging is provided by loading your own custom CSS style sheets. So, it can be argued that the frontend visualization of this system can be configured to almost any taste as needed and desired.

3. Intensity of project development.

If you look at the development of OJS, you can conclude that it is a project that is developing very rapidly and is supported by the developer community. Now two versions are offered for download - the LTS (Long-Term Support) version, which is characterized by guaranteed stability and is released approximately once every 3-4 months, and the version, which contains a greater number of innovations, but has a shorter support period, is released once 2-3 months (*Release Archive for OJS - Public Knowledge Project*, n. d.). Having two versions (LTS and ground-breaking version) allows users to choose between stability and innovation. This increases flexibility and allows users to choose the version that best suits their needs. Regular updates ensure that the system remains up-to-date and secure.

4. Quality of technical documentation.

The quality of the documentation is not very different from the documentation for any open-source project, in general it can be considered satisfactory. The main thing that causes confusion at first glance, as a clear example of a gap in the documentation, is the description of the installation process, which only talks about deploying the test system on the localhost. Not a word about how to do it on a real server and with the help of remote access. In addition, the options of the configuration file are not systematized, their description is scattered, situational and incomplete.

5. Software implementation.

The great advantage of OJS, in my opinion, is the support of the standard LAMP server stack, which reduces the system deployment process to a standard procedure that even students can perform in a few laboratory sessions (Ткачов, 2023, с. 81-85). The installation process is basically no different from installing the world's most popular content management system WordPress, which is used by more than 43% of sites.

As for the other platforms that make up the remaining 40% of open access journal distribution, here are the two most significant:

– ScholarOne is offered by Clarivate, a commercial journal publishing platform implemented by many prominent publishers whose journals are indexed in Web of Science. It offers a wide range of features, including tools for managing submissions, reviewing and publishing (*ScholarOne Journals Workflow Management Software | Clarivate,* n. d.);

– Editorial Manager, promoted by Elsevier, is another commercial journal publishing platform used by many publishers whose journals are indexed in Scopus. It is similar in functionality to ScholarOne but offers some unique features such as a plagiarism detection tool. Editorial Manager also integrates with many other systems (*Submission systems | Editor | Elsevier,* n. d.).

Discussion and Findings

This survey examines only a few of the many available platforms for open access repositories and journals. The best platform for you will depend on the goal of the project, specific needs and budget. When choosing a platform, the following should be considered first.

1. Your budget, which covers the following aspects:

– the cost of the system if you choose the commercial option;

– the cost, and related performance, of server equipment;

– the cost of maintaining your resource, in particular the salary costs associated with labor-intensive service processes;

– the cost of resource, at least for electricity, possibly also for the air conditioning and ventilation system of server rooms, if the placement of servers is not provided by the internet provider, and you do it at the expense of internal resources.

2. Your qualifications, or the qualifications of the personnel in the division of your institution, which deals with information technologies and ensuring the proper functioning of the network infrastructure.

3. Functionality of the implemented system in the context of your tasks and needs.

4. Ease of use and documentation. How easy is the platform to use? Will your researchers, editors, and reviewers be able to use it easily?

5. Support from developers. What level of support is offered for the platform? Are there resources available to help use the platform?

6. The possibility of integration with other systems, if any. Does the platform integrate with other systems you have implemented, such as your library system or research management system?

Top tip – do your research on different open access repository or journal platforms before choosing the one that's right for you. You'll be able to compare features, prices, and other factors on these platforms' websites, as well as read reviews on forums.

Practical recommendations

1. Validation of the correct operation of the OAI-PMH protocol.

Immediately after deploying the platform, even if you have not decided on the structure and way of presenting the data, follow this process. If you encounter OAI-PMH errors, you have only two alternatives - to look for ways to fix the errors or to choose another platform or another version of it. If there are OAI-PMH errors, then further debugging and deployment of the system does not make any sense.

OAI-PMH validation is performed using various online services, the most famous of which are:

– «OVAL :: BASE OAI-PMH Validator» (*OVAL :: BASE OAI-PMH Validator*, n. d.);

– «OAI-PMH Validator & data extractor» (*OAI-PMH Validator & data extractor - OAIPMH.com*, n. d.);

– «OpenAIRE's Repository Manager» (*OpenAIRE's Repository Manager*, n. d.);

– «OAI-PMH Data Provider Validation and Registration» (*OAI-PMH Data Provider Validation and Registration*, n. d.).

If you are planning to upgrade a production system to a new version, I would strongly recommend that you first install this upgrade on a test server and make sure that the OAI-PMH in the new version works and passes all validation tests.

2. System testing.

Conduct system testing in several directions, which will allow you to identify and eliminate possible problems at the early stages:

– find out if there are no critical system errors on the server during the launch of the main work processes;

– check the load on the server's processor, find out how many free resources remain after installing the system, and make appropriate conclusions about whether they are enough or whether a more productive server is needed;

– conduct testing with a large amount of data to determine how the system behaves under heavy loads, this will help identify the need for scaling;

– evaluate the speed of the system, in particular the speed of response to user requests and performance during operation, because now a long response time has begun to "annoy" search engine robots and is one of the reasons for poor content indexing.

3. Backup.

Find out exactly which data is critical and how you need to make backup copies that will ensure system recovery in the event of server equipment failure. Automate this process with command scripts to save time and reduce manual work. Determine how often you want to create backups. It can be daily, weekly or other periodicity, depending on the volume and frequency of changes in the information. Consider storing backups. It is best to have multiple copies on different media such as remote servers, external hard drives or cloud storage to protect against accidental deletion or corruption. Periodically test the backup restore procedure to ensure that it is working correctly. Testing will help to avoid unpleasant surprises in the event of a real need for recovery.

4. Development of detailed user and service documentation.

Develop detailed, illustrated, and most importantly, clear instructions for users. Train users, administrators, and technical staff. This will contribute to more efficient use of the system and reduce the number of operational errors. Keep the documentation current by regularly updating and adding new material, especially after new versions of the system or changes in functionality.

5. Moderation.

In fact, practical experience shows that no matter how detailed the user documentation is, how much time you devote to training users, all the "gaps" in which you cannot limit their actions by technical means turn out to be problematic over time. Only you understand that everything should be templated, uniform and presented according to certain norms, requirements or standards. Surprisingly, users do not want to understand this. So, without moderation, your project will slowly turn into a chaotic dump from the very beginning. It is good if this process does not affect the general operation of the platform, but the search function and the correctness of the information structure will be affected for sure.

Conclusions

1. The study expands the understanding of problems and challenges in the field of open science regarding the organizational and technical aspects of the deployment of platforms with the support of OAI-PMH.

2. The most popular platforms for deploying open access repositories and journals are considered, their advantages and disadvantages are shown.

3. The key errors and difficulties that organizers may face during the implementation of information resources with the support of OAI-PMH are identified, namely:

– incorrect implementation of the OAI-PMH protocol, the presence of errors in some specifications or functions of the protocol;

– lack of detailed system, technical and operational documentation;

– integration difficulties related to compatibility with other systems and platforms;

– lack of resources, both technical and financial, as well as human.

4. Based on the conducted research, practical recommendations for administrators and managers of open journals and repositories were developed. This information will also be useful when deploying open conferencing platforms. Practical approaches are proposed to ensure the reliable operation of the metadata collection protocol, initial debugging and further integration of these platforms.

5. The results of the study indicate that the successful operation of OAI-PMH requires not only a technical understanding of the protocol, but also a clear strategy and support at all levels of the organization. Technical operational errors can be minimized by developing detailed documentation and training personnel. Problems such as lack of resources require the involvement of management and the provision of adequate project financing.

6. The study is limited to case studies and does not cover all possible aspects of deployment and integration of OAI-PMH enabled platforms. Future research could focus on quantifying the implementation effectiveness of these platforms and their long-term impact on open science. Additionally, it is worth considering the influence of various social and economic factors on the success of OAI-PMH implementation.

References

1. *Open Archives Initiative*. (n. d.). Open Archives Initiative. https://openarchives.org/
2. *OpenDOAR Statistics - Sherpa Services*. (n. d.). Research Funders' Open Access Policies - Sherpa Services. https://v2.sherpa.ac.uk/view/repository_visualisations/1.html
3. *DSpace Home - DSpace*. (n. d.). DSpace. https://dspace.lyrasis.org/
4. *EPrints Services*. (n. d.). EPrints Services. https://www.eprints.org/uk/
5. *Features - DSpace*. (n. d.). DSpace. https://dspace.lyrasis.org/features
6. *Browse by Type - EPrints Files*. (n. d.). Welcome to EPrints Files - EPrints Files. https://files.eprints.org/view/type/translation.html
7. *History - EPrints Documentation*. (n. d.). EPrints Documentation. https://wiki.eprints.org/w/History
8. *Releases - DSpace - LYRASIS Wiki*. (n. d.). Dashboard - LYRASIS Wiki. https://wiki.lyrasis.org/display/DSPACE/Releases
9. *Installing DSpace - DSpace 7.x Documentation - LYRASIS Wiki*. (n. d.). Dashboard - LYRASIS Wiki. https://wiki.lyrasis.org/display/DSDOC7x/Installing+DSpace
10. *Category: Installation - EPrints Documentation*. (n. d.). EPrints Documentation. https://wiki.eprints.org/w/Category:Installation
11. *Что такое стек LAMP? – Объяснение стека LAMP – AWS*. (n. d.). Amazon Web Services, Inc. https://aws.amazon.com/ru/what-is/lamp-stack/
12. freeCodeCamp.org. (2020, 10 січня). *Interpreted vs Compiled Programming Languages: What's the Difference?* https://www.freecodecamp.org/news/compiled-versus-interpreted-

languages/

13. *Download*. (n. d.). WordPress.org. https://wordpress.org/download/

14. *Public Knowledge Project*. (n. d.). Public Knowledge Project. https://pkp.sfu.ca/

15. *Solution: Open Journal Systems (OJS) | Infra Finder*. (n. d.). Search & Compare Solutions | Infra Finder. https://infrafinder.investinopen.org/solutions/open-journal-systems-ojs

16. Ткачов, Ю. В. (2023). Вибір операційної системи для створення інформаційних ресурсів. *Світ наукових досліджень*, *17*, 81–85. https://doi.org/10.6084/m9.figshare.22433743

17. *ScholarOne Journals Workflow Management Software | Clarivate*. (n. d.). Clarivate. https://clarivate.com/products/scientific-and-academic-research/research-publishing-solutions/scholarone/

18. *Submission systems | Editors | Elsevier*. (n. d.). www.elsevier.com. https://www.elsevier.com/editor/submission-systems

19. *Open Journal Systems*. (n. d.). Weblate. https://translate.pkp.sfu.ca/projects/ojs/#languages

20. *Release Archive for OJS - Public Knowledge Project*. (n. d.). Public Knowledge Project. https://pkp.sfu.ca/software/ojs/download/archive/

21. *OVAL :: BASE OAI-PMH Validator*. (n. d.). OVAL :: BASE OAI-PMH Validator. http://oval.base-search.net/

22. *OAI-PMH Validator & data extractor - OAIPMH.com*. (n. d.). OAI-PMH Validator & data extractor - OAIPMH.com. https://validator.oaipmh.com/

23. *OpenAIRE's Repository Manager*. (n. d.). OpenAIRE's Repository Manager. https://provide.openaire.eu/home

24. *OAI-PMH Data Provider Validation and Registration*. (n. d.). Open Archives Initiative. https://www.openarchives.org/Register/ValidateSite