Sustainable Scholarly Publishing: Insights from DCPapers

Nishad Thalhath[1,*], Mitsuharu Nagamori[2] and Tetsuo Sakaguchi[2]

[1]Graduate School of Library, Information and Media Studies, University of Tsukuba, Tsukuba, Ibaraki, Japan
[2]Faculty of Library, Information and Media Studies, University of Tsukuba, Tsukuba, Ibaraki, Japan

Abstract

This paper examines the technological evolution and reimplementation of DCPapers, the open access publication platform for the Dublin Core Conference proceedings. Over two decades, DCPapers has served as the primary repository for metadata research and practice, publishing nearly twenty volumes that span theoretical frameworks from common purpose to post-structural approaches. While the platform has successfully facilitated knowledge dissemination through open access publication, the rapid advancement of web technologies and emerging security challenges have necessitated a comprehensive redesign. We present a systematic analysis of the platform's transformation, detailing the migration from a traditional content management system to a modern static publishing architecture. Our findings illuminate key challenges in maintaining long-term digital scholarly repositories and offer insights into the implementation of contemporary web technologies in academic publishing platforms. This work contributes to the broader discourse on sustainable open access publication systems and provides a practical framework for similar digital transformation initiatives in scholarly communication infrastructure.

Keywords

Digital Libraries, Open Access Publishing, Scholarly Communication, Static Site Generation, Web Architecture

1. Introduction

The Dublin Core Metadata Initiative (DCMI) has played a foundational role in shaping metadata standards and practices through its annual conferences, which serve as crucial forums for advancing information science discourse. Central to this contribution is DCPapers [1] , the initiative's open access publication platform that houses the conference proceedings. This platform represents a significant scholarly repository, comprising approximately twenty volumes that document the evolution of metadata practices, standards, and theoretical frameworks in the field.

The historical significance of DCPapers extends beyond its role as a mere repository. It stands as a chronicle of the metadata community's intellectual development, capturing the transition from early conceptual frameworks through to contemporary theoretical approaches.

DCMI-2024 International Conference on Dublin Core and Metadata Applications
*Corresponding author.
nishad@slis.tsukuba.ac.jp (N. Thalhath); nagamori@slis.tsukuba.ac.jp (M. Nagamori); saka@slis.tsukuba.ac.jp (T. Sakaguchi)
0000-0001-9845-9714 (N. Thalhath); 0000-0002-9545-7825 (M. Nagamori); 0000-0002-2055-5594 (T. Sakaguchi)

[1]https://dcpapers.dublincore.org/

https://doi.org/10.23106/dcmi.952446238

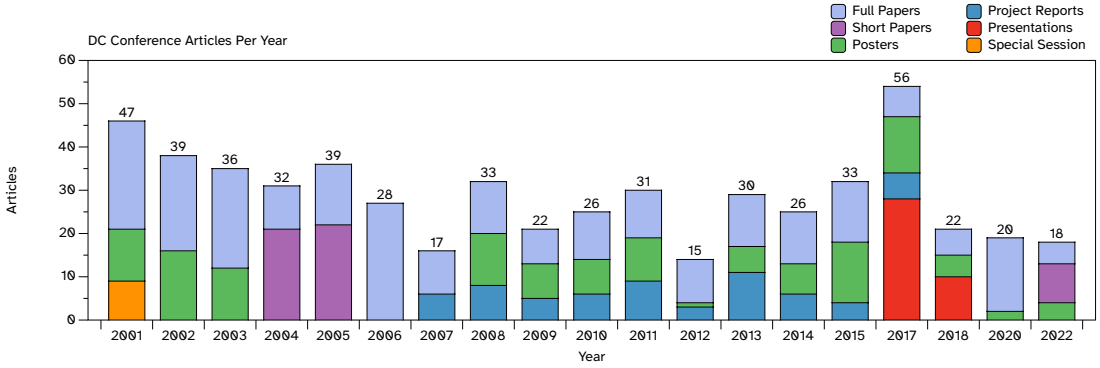

Figure 1: Distribution of articles by issue and type across conference proceedings

This evolution is reflected in the platform's content, which spans from foundational discussions of common purpose to more nuanced explorations of post-structural metadata theory.

1.1. Significance of DCPapers

The Dublin Core Conference series occupies a unique position within the metadata community, distinguished by its innovative hybrid model that bridges academic research and practitioner expertise. This distinctive approach has fostered the development of the Dublin Core papers, which synthesize theoretical insights with practical applications. The resulting body of work provides comprehensive perspectives on metadata practices and their evolution, serving both scholarly and professional audiences.

The proceedings' significance is evidenced by their substantial citation impact and continued relevance to contemporary metadata discourse. These papers have not only established fundamental concepts but have also demonstrated remarkable resilience in their applicability to emerging metadata challenges. Their enduring influence stems from rigorous examination of both theoretical principles and practical implementations, creating a valuable resource for researchers and practitioners alike.

1.2. DCEvents and Platform Integration

A critical component of the DCMI's digital infrastructure is the DCEvents portal [2] , which operates in concert with DCPapers to create a seamless workflow from event organization through to publication. DCEvents utilizes PKP's Open Conference Systems (OCS) [3] to archive and manage community activities, while DCPapers employs PKP's Open Journal Systems (OJS) [4]for publishing proceedings. This integrated approach was designed with a strong emphasis on interoperability, facilitating efficient transition from submission through to final publication.

[2]https://dcevents.dublincore.org/
[3]https://pkp.sfu.ca/software/retired/
[4]https://pkp.sfu.ca/software/ojs/

https://doi.org/10.23106/dcmi.952446238

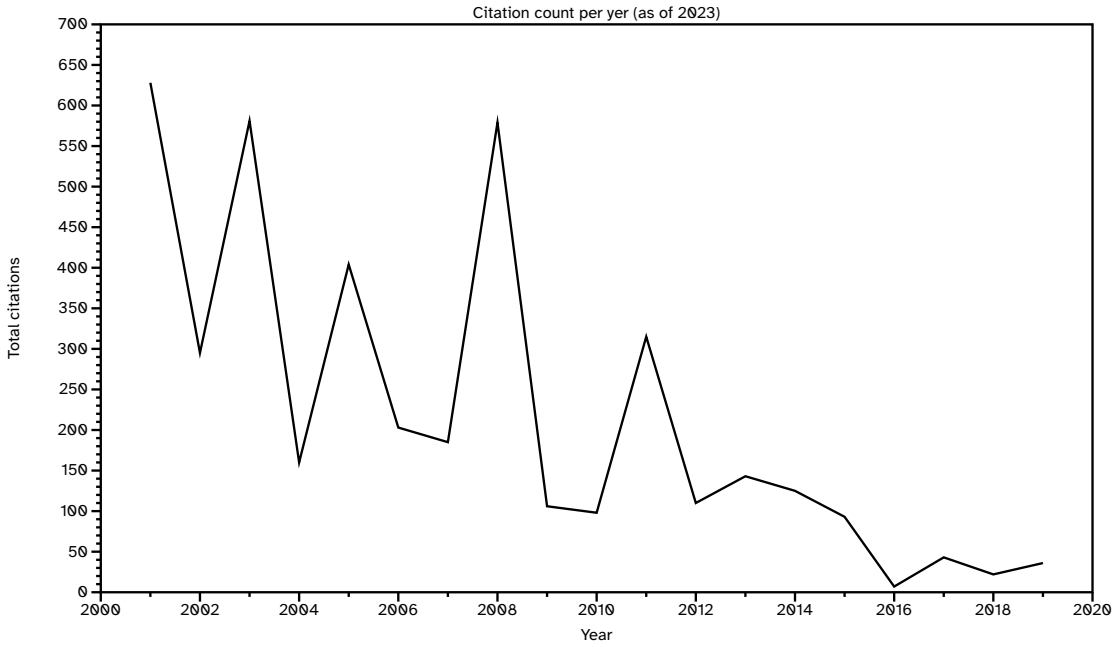

Figure 2: Annual citation metrics for Dublin Core Conference proceedings

However, the technological landscape has evolved significantly since DCPapers' inception. Contemporary web technologies, security requirements, and user expectations have advanced beyond the capabilities of traditional content management systems. This evolution, coupled with persistent security challenges and maintenance complexities, has necessitated a fundamental redesign of the DCPapers platform.

This paper documents our experience in reimagining and reconstructing DCPapers for the modern web environment. We present a systematic analysis of the challenges encountered, solutions implemented, and lessons learned throughout this transformation process. Our findings contribute to the broader discourse on sustainable scholarly communication infrastructure and provide practical insights for similar digital platform modernization initiatives.

1.3. PKP OJS Architecture and Limitations

The Public Knowledge Project's Open Journal Systems (PKP OJS) architecture, while widely adopted in academic publishing, presents significant operational challenges in contemporary web environments. Built on a traditional PHP, MySQL, and Apache stack, the system requires comprehensive server infrastructure that introduces complexities in maintenance, security, and long-term sustainability.

https://doi.org/10.23106/dcmi.952446238

1.3.1. Security Vulnerabilities

Security analysis of the PKP OJS implementation revealed persistent vulnerabilities that pose significant risks to scholarly content. The system has demonstrated susceptibility to various attack vectors, including cross-site scripting (XSS), SQL injection, and unauthorized access attempts. Despite regular security patches and updates, the fundamental architecture remains vulnerable to emerging cyber threats, necessitating constant vigilance and resource-intensive monitoring.

The Dublin Core website and its associated web resources have experienced frequent security incidents, ranging from automated breach attempts to targeted attacks. This ongoing security pressure has highlighted the limitations of maintaining a dynamic, database-driven system in an increasingly hostile online environment. These challenges ultimately catalyzed the decision to transition toward a more secure, read-only architecture.

1.3.2. Technical Debt and Maintenance Overhead

The PKP OJS infrastructure demands continuous maintenance to ensure compatibility with evolving web standards and security requirements. The system's reliance on PHP and MySQL necessitates regular version updates, security patches, and compatibility testing. This maintenance burden creates significant technical debt, requiring substantial resources for routine operations rather than platform enhancement or content development.

Furthermore, the Apache server configuration, while robust, requires extensive optimization to maintain acceptable performance levels under varying traffic conditions. The complexity of this stack increases the potential for misconfiguration and creates additional attack surfaces that must be actively monitored and managed.

1.3.3. Scalability and Performance Constraints

Performance analysis has revealed inherent limitations in the PKP OJS architecture's ability to scale efficiently. The system's dynamic content generation and database-dependent operations can lead to degraded performance under high load conditions, particularly during peak usage periods such as conference submissions and publication releases. These scalability constraints impact user experience and system reliability, potentially limiting the platform's ability to serve its growing user base effectively.

The platform's role-based access control system, while comprehensive, introduces additional complexity to the system architecture. The intricate workflow management features, designed to facilitate academic publishing processes, often require extensive customization and can create bottlenecks in content processing and publication workflows.

1.3.4. Integration and Interoperability Challenges

PKP OJS provides robust support for established scholarly publishing standards and protocols, including OAI-PMH, DOI integration, and SWORD compliance. However, as the scholarly communication landscape evolves, the system faces certain challenges in adapting to emerging requirements and technologies. While its core functionality supports traditional scholarly

https://doi.org/10.23106/dcmi.952446238

workflows effectively, the architectural constraints of OJS can present obstacles when implementing newer standards or integrating with modern research information systems and metadata aggregators.

These integration limitations primarily manifest in scenarios requiring real-time data exchange, complex API interactions, or integration with contemporary preservation services. The system's underlying architecture, though reliable for traditional publishing workflows, requires significant development effort to accommodate rapidly evolving scholarly communication practices and emerging standards in research information management.

The combination of these technical, security, and operational challenges has influenced our decision to explore alternative architectural approaches for the DCPapers platform. This decision reflects not a shortcoming of OJS itself, but rather the specific requirements of our use case and the evolving needs of our scholarly community.

2. Static Publishing

Static publishing architectures fundamentally differ from traditional dynamic content management systems by serving pre-rendered content rather than generating pages on demand. In this paradigm, web pages are pre-built during a compilation phase and served directly to users without database queries or server-side processing. This approach yields significant advantages in performance, security, and reliability while introducing distinct constraints that must be carefully considered in scholarly publishing contexts [1].

The static publishing model aligns particularly well with scholarly content, which typically maintains stability post-publication. By leveraging modern static site generation technologies, platforms can achieve optimal performance characteristics while significantly reducing infrastructure complexity and security vulnerabilities.

2.1. Technical Implications and Constraints

Content Management Workflows. The transition from dynamic to static publishing necessitates a fundamental reimagining of content management workflows. Unlike traditional content management systems that provide real-time editing and preview capabilities, static publishing requires a more structured approach to content updates. Changes must proceed through a defined build and deployment pipeline, introducing additional complexity in the editorial process.

Performance Considerations. Static publishing excels in content delivery performance, as pre-rendered pages can be served directly from edge networks without database queries or server-side processing. However, this approach requires careful attention to build optimization and asset management, particularly when handling large volumes of scholarly content. The build process must be efficiently orchestrated to maintain reasonable deployment times while ensuring content integrity.

Search and Discovery. The implementation of search functionality in static environments presents unique challenges. Without server-side query processing, alternative approaches such

https://doi.org/10.23106/dcmi.952446238

as client-side search indexes or external search services must be carefully integrated to maintain robust discovery capabilities. These solutions must be optimized to handle scholarly-specific requirements, including complex metadata queries and citation networks.

2.2. Migration Challenges

URL Preservation. Maintaining persistent URLs is crucial in scholarly publishing to preserve citation integrity and ensure continued access to research materials. The migration process must carefully map existing URL structures to their static equivalents while implementing appropriate redirect mechanisms for legacy URLs. This preservation of scholarly links requires meticulous attention to URL routing and redirect configuration.

Metadata Management. The transition of dynamic metadata systems to static alternatives requires careful consideration of data modeling and access patterns. While dynamic systems often generate metadata on demand, static publishing necessitates pre-computation of all possible metadata views and relationships. This requirement influences both the build process and the overall information architecture of the platform.

Content Transformation. The conversion of content from dynamic to static formats involves complex transformation processes. These include:

- Converting dynamic database queries to static data structures
- Implementing efficient build processes for large content repositories
- Preserving rich metadata relationships in static contexts
- Maintaining content integrity throughout the transformation process

These challenges must be addressed while ensuring the resulting system maintains the accessibility, discoverability, and usability expected of modern scholarly platforms.

3. Alternative Publishing Platforms

The evolution of digital scholarly publishing has produced various platforms that address different aspects of academic content management and dissemination. This section examines key alternatives that were evaluated during the DCPapers redesign process, analyzing their strengths and limitations within the context of scholarly publishing requirements.

3.1. Static Site Generation Platforms

Hugo Framework. Hugo [5] represents a significant advancement in static site generation technology, offering exceptional build performance and sophisticated content management capabilities. Built on Go, it provides a robust foundation for handling large-scale documentation and academic content. Its key strengths include instantaneous builds, flexible taxonomy systems, and comprehensive template capabilities. However, its learning curve and specific content organization requirements necessitate careful consideration in academic publishing contexts.

[5] https://gohugo.io/

https://doi.org/10.23106/dcmi.952446238

Jekyll. While not explicitly evaluated for DCPapers, Jekyll's [6] widespread adoption in academic publishing merits discussion. Its integration with GitHub Pages and extensive academic templates has established it as a popular choice for scholarly websites. However, its build performance limitations with large content repositories make it less suitable for comprehensive conference proceedings platforms.

3.2. Institutional Repository Platforms

DSpace Platform. DSpace [7] stands as a comprehensive solution for institutional repositories, offering robust digital asset management capabilities and established workflows for academic content. Its strengths lie in standardized metadata handling, OAI-PMH compliance, and preservation features. The platform supports various content types and implements granular access controls. However, its monolithic architecture and significant infrastructure requirements present challenges for lighter-weight publishing needs.

EPrints System. EPrints[8] provides specialized functionality for building open access repositories with strong emphasis on bibliographic data management. Its architecture facilitates efficient handling of research outputs and supports extensive metadata customization. While powerful for institutional repositories, its complexity and maintenance requirements may exceed the needs of conference-specific publishing platforms.

Omeka Platform. Omeka[9] distinguishes itself through its focus on digital exhibitions and collections, offering sophisticated presentation capabilities for scholarly materials. Its emphasis on user-friendly content management and exhibition tools makes it particularly suitable for curated academic collections. However, its architecture may not optimally support the specific workflows required for conference proceedings publication.

3.3. Contemporary Frameworks

Recent developments in web technologies have introduced new possibilities for scholarly publishing platforms. Modern JavaScript frameworks and JAMstack architectures offer promising alternatives for building efficient, secure, and maintainable publishing systems. These approaches combine the benefits of static site generation with dynamic capabilities through API integration, potentially addressing many limitations of traditional publishing platforms.

The evaluation of these alternatives informed our architectural decisions for the DCPapers redesign, highlighting the need to balance modern web development practices with the specific requirements of scholarly publishing. This analysis contributed to our ultimate decision to implement a custom solution incorporating elements from multiple approaches while maintaining focus on the unique needs of the Dublin Core community.

[6] https://jekyllrb.com/
[7] https://dspace.org/
[8] https://www.eprints.org/
[9] https://omeka.org/

https://doi.org/10.23106/dcmi.952446238

4. Evolution of Web Publishing Technologies

The landscape of web publishing continues to undergo significant transformation, driven by advances in distributed computing, edge delivery networks, and modern JavaScript ecosystems. These technological developments present new opportunities for scholarly publishing platforms while introducing important considerations for implementation.

4.1. Serverless Architectures

The emergence of serverless computing represents a paradigm shift in web application deployment and scaling. This approach eliminates traditional server management concerns while offering significant advantages for scholarly publishing. Serverless architectures provide automatic scaling based on demand, efficiently handling varying loads common in academic publishing cycles. The consumption-based pricing model aligns particularly well with academic publishing workloads, potentially reducing operational costs compared to traditional hosting solutions.

4.2. Edge Computing Paradigm

Edge computing fundamentally transforms content delivery by processing data closer to its consumption point [2]. For scholarly publishing platforms, this distribution is crucial for international academic communities, ensuring consistent access speeds regardless of user location. Advanced caching at edge locations enables optimal delivery of scholarly content, particularly beneficial for static assets such as PDFs and supplementary materials, significantly reducing latency while maintaining content integrity.

4.3. Modern JavaScript Ecosystem

The evolution of JavaScript and ECMAScript standards continues to enhance web application capabilities. Standardized web components enable the creation of reusable, encapsulated UI elements specific to scholarly content presentation, facilitating consistent rendering of complex academic content. Modern JavaScript frameworks provide sophisticated optimization techniques, including tree-shaking, code splitting, and lazy loading, enabling the delivery of rich interactive features while maintaining optimal performance characteristics.

4.4. Read-Only Architecture

The concept of read-only web architectures has gained prominence as a response to security challenges and maintenance complexity. This approach significantly reduces attack surfaces by eliminating dynamic content generation and database dependencies, particularly relevant for scholarly content that typically remains static post-publication. The simplification of infrastructure requirements reduces operational complexity and maintenance overhead, allowing organizations to focus resources on content quality and platform improvements. Read-only architectures provide robust guarantees for content immutability, ensuring that published research remains unchanged and verifiable over time.

https://doi.org/10.23106/dcmi.952446238

These technological trends collectively influence the development of modern scholarly publishing platforms, offering new possibilities for improving content delivery, reducing operational complexity, and enhancing security. The integration of these technologies must be carefully considered within the context of academic publishing requirements, ensuring that adoption of new approaches aligns with the fundamental goals of scholarly communication.

5. Platform Design and Implementation

The redesign of DCPapers required careful consideration of modern web technologies and development frameworks while maintaining focus on scholarly publishing requirements. Our implementation leverages contemporary tools and approaches to create a robust foundation for academic content delivery.

5.1. SvelteKit Framework Implementation

SvelteKit [10] was selected as the primary framework for the DCPapers redesign, offering significant advantages for scholarly content delivery. Its compile-time approach to JavaScript optimization results in minimal runtime overhead, crucial for delivering academic content efficiently. The framework provides a modern development environment with filesystem-based routing and comprehensive build optimization, facilitating maintainable code organization while supporting complex routing requirements. Additionally, SvelteKit's intelligent client-side hydration strategy ensures optimal performance across diverse device capabilities and network conditions, particularly valuable for global academic communities accessing content through varying infrastructure quality.

5.2. Data Management Architecture

The implementation leverages TOML (Tom's Obvious, Minimal Language)[11] for structured data management, offering clear semantic structure for metadata representation and robust parsing with strict type checking. This choice facilitates direct content editing while remaining version control friendly, essential for maintaining scholarly content integrity.

The platform implements a Git-based content management workflow, providing comprehensive version control and transparent audit trails of editorial modifications. This approach enables collaborative review processes through pull requests and automated deployment triggers for content updates. Integration with Cloudflare's infrastructure enables automated build and deployment processes, global content distribution through edge networks, and robust security protections [3].

5.3. Semantic Web Integration

The platform implements comprehensive support for linked data standards and semantic web technologies. Rich structural metadata is embedded using JSON-LD, facilitating machine-

[10] https://svelte.dev/
[11] https://toml.io/en/

https://doi.org/10.23106/dcmi.952446238

readable semantic descriptions and integration with scholarly indexing services. The integration of RDFa markup provides in-line semantic annotation of content and granular metadata attribution, supporting citation networks and enhancing accessibility of scholarly metadata.

The system maintains robust connections with external scholarly systems through Wikidata entity alignment, Semantic Scholar article matching, and integration with ORCID author identifiers and DOI resolution services. This comprehensive approach ensures that DCPapers meets modern web standards while maintaining the specific requirements of scholarly publishing platforms, creating a robust foundation for future development.

6. Design Principles and Standards

The redesign of DCPapers was guided by established principles in open access scholarly communication and data management [4]. While not all elements could be fully implemented in the initial release, these principles provided a framework for development decisions and future platform evolution.

6.1. Budapest Open Access Initiative as a Core Model

The Budapest Open Access Initiative (BOAI) [12] serves as a foundational framework for the platform's approach to scholarly content dissemination. The initiative's core principles of unrestricted access and reuse rights inform key aspects of the platform architecture. By adopting these principles, DCPapers aims to facilitate broad dissemination of scholarly content while maintaining appropriate attribution and rights management. [5]

The platform's design reflects BOAI's emphasis on removing barriers to scholarly content access [6]. This approach extends beyond basic availability to consider various aspects of content accessibility, including machine readability and format flexibility. While complete implementation of all BOAI recommendations remains an ongoing process, these principles continue to guide platform development decisions.

6.2. Adapting the FAIR Model to Scholarly Publishing

The FAIR principles (Findable, Accessible, Interoperable, and Reusable) [7], originally developed for research data management, provide valuable guidance for scholarly publishing platforms. DCPapers adapts these principles to the specific context of conference proceedings publication:

Findability Goals. The platform architecture prioritizes content discoverability through persistent identifiers, rich metadata, and integration with scholarly indexes. This includes implementation of DOIs and structured metadata schemas, with provisions for expanding discovery mechanisms in future iterations.

Accessibility Considerations. Platform design emphasizes reliable content access through standard protocols and interfaces. While technical accessibility forms the foundation, ongoing work continues to enhance accessibility across different user contexts and needs [8].

[12] https://www.budapestopenaccessinitiative.org

https://doi.org/10.23106/dcmi.952446238

Interoperability Framework. The architecture supports interoperability through standard metadata schemas and identifier systems. Initial implementation focuses on core scholarly communication protocols, with flexibility for expanding integration capabilities as community needs evolve.

Reusability Support. The platform emphasizes clear licensing information and structured metadata to support content reuse. This approach facilitates both human and machine consumption of scholarly content while respecting attribution requirements.

6.3. Future Directions

The development of DCPapers represents an ongoing commitment to improving scholarly communication infrastructure. A primary objective is the release of the platform as open-source software, accompanied by comprehensive documentation to facilitate adoption and community contributions. This initiative aims to promote transparency in scholarly publishing infrastructure while supporting customization for diverse publishing contexts.

The implementation of Dublin Core's Core Application Profile (DCSRAP) [13] represents another significant planned enhancement. This structured metadata model will enhance the representation of scholarly relationships and improve interoperability with external systems, supporting emerging requirements in the scholarly communications landscape.

7. Limitations

While the DCPapers platform successfully addresses many challenges in scholarly publishing, several limitations warrant acknowledgment. The adoption of a static publishing approach, while beneficial for security and performance, introduces constraints on real-time content updates and dynamic features. Large repositories of scholarly content can lead to extended build times, requiring careful consideration in content management workflows.

The reliance on SvelteKit, while advantageous for development efficiency, introduces potential sustainability considerations due to the framework's relative novelty in the ecosystem. Additionally, the transition from traditional content management systems presents ongoing challenges in maintaining legacy content relationships and ensuring consistent content representation.

The implementation of planned enhancements requires sustained development effort, which may be constrained by available resources within the academic community. However, these acknowledged constraints inform ongoing development priorities while the platform's core functionality effectively serves its primary purpose of facilitating open access to Dublin Core conference proceedings.

8. Conclusion

The transformation of the DCPapers publishing platform represents a significant advancement in open-access scholarly communication infrastructure. Through this redesign process, we have

[13] https://dcmi.github.io/dc-srap/

https://doi.org/10.23106/dcmi.952446238

demonstrated the viability of modern web technologies and static publishing approaches in meeting the evolving needs of the metadata community while addressing persistent challenges in academic publishing platforms.

The adoption of contemporary web frameworks and static site generation techniques has enabled significant improvements in platform security, performance, and maintainability. These technological choices, guided by established principles from the Budapest Open Access Initiative and FAIR guidelines, have resulted in a more robust and sustainable publishing infrastructure. The integration of structured metadata approaches and semantic web technologies ensures that the platform supports both traditional scholarly communication needs and emerging requirements for machine-readable academic content.

The challenges encountered during this transformation, particularly in areas of content migration and workflow adaptation, have provided valuable insights into the complexities of modernizing established scholarly publishing systems. These experiences contribute to the broader discourse on sustainable academic infrastructure and offer practical guidance for similar digital transformation initiatives.

Looking forward, the DCPapers platform's open-source trajectory and planned metadata enhancements position it to continue serving the evolving needs of the Dublin Core community while contributing to the advancement of open scholarly communication practices. The lessons learned through this redevelopment process highlight both the opportunities and challenges in modernizing academic publishing infrastructure, providing a foundation for future innovations in this domain.

References

[1] D. Markovic, M. Scekic, A. Bucaioni, A. Cicchetti, Could jamstack be the future of web applications architecture? an empirical study, in: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, SAC '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 1872–1881. URL: https://doi.org/10.1145/3477314.3506991. doi:10.1145/3477314.3506991.

[2] N. Kamiyama, Y. Nakano, K. Shiomoto, G. Hasegawa, M. Murata, H. Miyahara, Analyzing effect of edge computing on reduction of web response time, in: 2016 IEEE Global Communications Conference (GLOBECOM), 2016, pp. 1–6. doi:10.1109/GLOCOM.2016.7841607.

[3] D. B. M. S.-V. J. D. Evan Peter Williamson, Olivia M. Wikle, J. Martinez, Using static web technologies and git-based workflows to re-design and maintain a library website (quickly) with non-technical staff, College & Undergraduate Libraries 28 (2021) 129–147. URL: https://doi.org/10.1080/10691316.2021.1887036. doi:10.1080/10691316.2021.1887036. arXiv:https://doi.org/10.1080/10691316.2021.1887036.

[4] H. Van De Sompel, C. Lagoze, The Santa Fe Convention of the Open Archives Initiative, D-Lib Magazine 6 (2000). URL: http://www.dlib.org/dlib/february00/vandesompel-oai/02vandesompel-oai.html. doi:10.1045/february2000-vandesompel-oai.

[5] J. Frank, R. Foster, C. Pagliari, Open access publishing – noble intention, flawed reality, Social Science & Medicine 317 (2023) 115592. URL: https://www.sciencedirect.com/science/article/pii/S027795362200898X. doi:10.1016/j.socscimed.2022.115592.

[6] D. Kwon, More than 100 scientific journals have disappeared from the Internet, Nature (2020). URL: https://www.nature.com/articles/d41586-020-02610-z. doi:10.1038/d41586-020-02610-z.

[7] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons, The FAIR Guiding Principles for scientific data management and stewardship, Scientific Data 3 (2016) 160018. URL: https://www.nature.com/articles/sdata201618. doi:10.1038/sdata.2016.18.

[8] S. Wild, Millions of research papers at risk of disappearing from the Internet, Nature 627 (2024) 256–256. URL: https://www.nature.com/articles/d41586-024-00616-5. doi:10.1038/d41586-024-00616-5.

Declaration of Generative AI and AI-Assisted Technologies in the Preparation of This Article: In preparing this Article, the authors utilized Grammarly, Writefull, and the OpenAI APIs to enhance readability, correct grammar, and refine language use. After employing these services, the authors thoroughly reviewed and edited the content as necessary and took full responsibility for this article's content.

https://doi.org/10.23106/dcmi.952446238