YADDA2 – Assemble Your Own Digital Library Application from Lego Bricks

Wojtek Sylwestrzak
Centre for Open Science
ICM, Univ. of Warsaw
ul. Prosta 69
00-838 Warszawa, Poland
w.sylwestrzak@icm.edu.pl

Tomasz Rosiek
Centre for Open Science
ICM, Univ. of Warsaw
ul. Prosta 69
00-838 Warszawa, Poland
t.rosiek@icm.edu.pl

Łukasz Bolikowski
Centre for Open Science
ICM, Univ. of Warsaw
ul. Prosta 69
00-838 Warszawa, Poland
l.bolikowski@icm.edu.pl

ABSTRACT

YADDA2 is an open software platform which facilitates creation of digital library applications. It consists of versatile building blocks providing, among others: storage, relational and full-text indexing, process management, and asynchronous communication. Its loosely-coupled service-oriented architecture enables deployment of highly-scalable, distributed systems.

Categories and Subject Descriptors

H.3.7 [**Information Storage and Retrieval**]: Digital Libraries

General Terms

Design, Performance

1. INTRODUCTION

The document presents the recent results in development of an open, flexible, high-performance software platform for digital library applications, which we named YADDA2.

1.1 Motivation

Since the late 1990s, Interdisciplinary Centre for Mathematical and Computational Modelling (ICM) at the University of Warsaw has been providing access to full texts of major scientific publishers (Elsevier, Springer, and recently IEEE) for the Polish scientific community. We realized very quickly that maintaining individual publishers' platforms is expensive and cumbersome, and we started building our own software, named YADDA, to provide a full text search engine and a single point of access to the heterogeneous content. Components of the YADDA software found their way into a number of other digital library systems, including D-NET (the DRIVER Network Evolution Toolkit software suite [1]) or EuDML (the European Digital Mathematics Library [7]). As both the software and the collections of documents grew, the original design could no longer meet our needs, especially in terms of flexibility and scalability. Based on lessons learned and careful examination of other available solutions, we have designed and are currently developing a modular software platform named YADDA2, which could then be used to build concrete DL applications by us and by others alike.

JCDL'12, June 10–14, 2012, Washington, DC, USA.
ACM 978-1-4503-1154-0/12/06.

1.2 Related work

There are already several products and software frameworks, often mature, often distributed under open-source licenses, which boast high flexibility, modularity, and ability to work with third-party systems. Notable examples include: aDORe federation architecture [8], capable of storing hundreds of millions of digital objects and terabytes of image files; CiteSeerX [4], a database of research publications in computer and information science and related areas; dLibra Digital Library Framework [6] developed at Poznan Supercomputing and Networking Center; D-NET (DRIVER Network Evolution Toolkit) [5], originally created for the DRIVER repository infrastructure; Greenstone [10, 9], produced by the New Zealand Digital Library Project at the University of Waikato; H2O platform infrastructure, launched by HighWire Press[1]; INVENIO[2], originally developed at CERN; NCore [2], powered by Fedora [3], a framework for creation, management, and preservation of digital content.

2. SOFTWARE PLATFORM

2.1 Design goals

Design of the YADDA2 software platform was driven by a need for high flexibility and scalability. We needed a software platform that would facilitate creation of several types of products: *stand-alone repositories* with a web front-end and a publishing application in the back-end; *repository federations* containing of multiple autonomous collections, accessed though a central front-end; *publication data warehouses* aggregating content from multiple repositories in order to provide long-term preservation of data and access for researchers and analysts.

Ideally, building a new product should be reduced to assembling reusable, configurable components. Generic services such as metadata and content storages, full-text index, relational index, batch processing engine, or authorization and authentication should be readily available, their configuration and assembly should be straightforward. At the same time, an option to build custom compontents (in various programming languages) and bridges should be retained.

[1]See: http://highwire.stanford.edu/publishers/H2O.dtl
[2]See: http://invenio-software.org/

The platform should be able to handle objects of various types (documents, data sets, audio-visual content) and in various formats. Typically, components should be prepared to handle tens or hundreds of millions of objects. It should be easy to deploy distributed systems in a way that would be transparent to the individual components. Finally, the platform should seamlessly interoperate with third-party systems by embracing open protocols and standards for information exchange.

2.2 Architecture

YADDA2 allows to build distributed heterogeneous systems with multi-layer architecture. In most cases, an architecture of a YADDA2 based system would consist of two tiers: the base services tier and the applications tier. The base services provide generic functionalities which are independent of the type of content being stored or otherwise processed. The applications, on the other hand, use base services to provide business logic and user interface.

YADDA2 architecture was designed in order to provide high performance and scalability, with Service-Oriented Architecture making it easy to seamlessly plug in additional processing resources or mass storage to existing production environments. At the same time it offers the flexibility of open architecture. Users of the platform can easily add new services or adapt applications to their specific needs. One of the main features of the YADDA2 architecture is its ability to be deployed it in distributed environment, spreading across multiple organizations with different licencing and authentication policies. The platform satisfies the requirement of flexible maintenance and security issues management in infrastructure managed by more than one institution. YADDA2 includes advanced security context management tools and allows to manage either fine-gained security policies within particular application or coarse-grained licensing and authentication policies related to the access to particular services by particular institutions. The ability to effectively access the platform's resources not only with front-end applications but also with internal APIs, allows to easily create ad-hoc analysis and data post-processing tools by researchers.

From the technical point of view, YADDA2 architecture identifies the following components comprising the hosting platform: *hosting infrastructure* – service registries and service containers, responsible for instantiating, managing and communication between individual services; *core services* – components providing particular low level aspects of the platform's functionality (the current version of the platform includes Metadata and Content Storage, Full-text Index, Similarity Index, Batch Processing Engine, Relational Index, and User Annotation Service); and *platform clients* – software components using the services of the infrastructure.

YADDA2 is based on the Java language and supports a number of communication standards including HTTP, REST, SOAP and RMI. In addition, applications created on a base of YADDA2 support interfaces specific to digital libraries like OAI-PMH and OpenSearch.

3. SUMMARY

We have presented YADDA2, a new software platform for digital libraries. One differentiating factor is its open, modular, loosely-coupled design. Another one is its scalability, not only in the sense of large amounts of data or large traffic volumes that the derived applications can reliably handle, but also its multi-scale capabilities, making it particularly easy to tailor the size of an application to the specific needs, from a small, local (or embedded) repository implementation, to large scale, complex distributed systems handling heterogeneous content and services.

4. ACKNOWLEDGEMENTS

This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information".

5. REFERENCES

[1] M. Artini, L. Candela, D. Castelli, P. Manghi, M. Mikulicic, and P. Pagano. Sustainable Digital Library Systems over the DRIVER Repository Infrastructure. *Lecture Notes in Computer Science*, 5173:227–231, 2008.

[2] D. B. Krafft, A. Birkland, and E. J. Cramer. Ncore: architecture and implementation of a flexible, collaborative digital library. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries - JCDL '08*, page 313, New York, New York, USA, 2008. ACM Press.

[3] C. Lagoze, S. Payette, E. Shin, and C. Wilper. Fedora: an architecture for complex objects and their relationships. *International Journal on Digital Libraries*, 6(2):124–138, Dec. 2005.

[4] H. Li, I. Councill, W.-C. Lee, and C. L. Giles. CiteSeerx: An architecture and Web service design for an academic document search engine. In *Proceedings of the 15th international conference on World Wide Web - WWW '06*, page 883, New York, New York, USA, 2006. ACM Press.

[5] P. Manghi, M. Mikulicic, L. Candela, D. Castelli, and P. Pagano. Realizing and Maintaining Aggregative Digital Library Systems: D-NET Software Toolkit and OAIster System. *D-Lib Magazine*, 16(3/4), Mar. 2010.

[6] C. Mazurek, T. Parkoła, and M. Werla. Distributed Digital Libraries Platform in the PIONIER Network. *Lecture Notes in Computer Science*, 4172:488–491, 2006.

[7] W. Sylwestrzak, J. Borbinha, T. Bouche, A. Nowiński, and P. Sojka. EuDML—Towards the European Digital Mathematics Library. In *Towards a Digital Mathematics Library*, pages 11–26, 2010.

[8] H. Van de Sompel, R. Chute, and P. Hochstenbach. The aDORe federation architecture: digital repositories at scale. *International Journal on Digital Libraries*, 9(2):83–100, Oct. 2008.

[9] I. H. Witten, D. Bainbridge, and D. M. Nichols. *How to Build a Digital Library*. Morgan Kaufmann, 2nd edition, 2009.

[10] I. H. Witten, S. J. Boddie, D. Bainbridge, and R. J. McNab. Greenstone: a comprehensive open-source digital library software system. In *Proceedings of the fifth ACM conference on Digital libraries - DL '00*, pages 113–121, New York, New York, USA, 2000. ACM Press.