Information Services & Use 34 (2014) 297–307
DOI 10.3233/ISU-140750
IOS Press

Shared service components infrastructure for enriching the user experience in electronic publications

Nikos Houssos *, Panagiotis Stathopoulos, Ioanna-Ourania Stathopoulou, Andreas Kalaitzis and Alexandros Soumplis

National Documentation Centre, National Hellenic Research Foundation, Athens, Greece

Abstract. A major requirement for electronic publishing systems is the availability of rich and intuitive mechanisms that enhance the user experience of viewing and searching online electronic documents such as books, monographs and journal papers. This work concerns a set of infrastructural components and their utilization for the creation of related coherent services and features for end users. We present a set of sophisticated platforms, tools and mechanisms that have been employed in real-life cases for implementing document viewing and full-text search features, shared among application instances of various types. Challenges encountered and the provided solutions are discussed.

Keywords: Electronic publishing, user experience, document viewers, online readers, image servers, JPEG2000, full-text search, optical character recognition

1. Introduction

Providing users with a rich, intuitive and meaningful interface for accessing online electronic resources, e.g. books, monographs, journal papers is a significant factor for improving the user experience in electronic publishing systems and consequently for their increased adoption and usage. Traditionally, electronic journals, and scientific publications, have been published and viewed online, using the widespread PDF format, due to the rigid but portable formatting it provides, and HTML for versatility.

Recently, initiatives such as the "Google Books" and "Google Art" project, the Internet Archive "Open Library" and advanced repository systems, have paved the way for novel online reading capabilities and experience, with features such as "page by page" viewing of electronic resources and tile-based image viewing systems, exploiting advanced codes such as JP2000 and corresponding online viewers.

These advances are becoming gradually available in electronic publication systems [1] and, if incorporated in Open Access systems (e.g. journals, repositories) can contribute to even wider adoption by users and publishers. Such incorporation will enable, apart from intuitive reading capabilities, the efficient viewing of large data sets visualisations, maps and/or images of cultural artifacts.

In this contribution, we present the technologies, mechanisms and open source components employed to achieve this functionality. The corresponding infrastructure and tools includes, among others, an interactive online reader with dynamic zooming, thumbnail view, full-text search and hit highlighting

[*] Corresponding author. E-mail: nhoussos@ekt.gr.

capabilities, a JPEG2000 image server, a highly scalable back-end common infrastructure for storage, batch image processing, OCR, indexing/search and access to digital files by multiple applications.

The rest of the article has the following structure: Section 2 describes the environment of real-life services and applications where the solutions proposed in this article have been applied. Section 3 elaborates on the development of advanced document viewing user experience based on a range of back-end and front-end services. Section 4 concerns the support of full-text search in various environments. The paper ends with a summary section.

2. Services and applications context

This section briefly presents the environment of real world services and applications where the solutions proposed in this article have applied. Essentially, we have implemented them in the context of scholarly communications infrastructure and in particular multiple epublishing systems and repositories that are operated by the National Documentation Centre of Greece (EKT).

2.1. The EKT electronic publishing platform

The National Documentation Centre of Greece has launched its own electronic publishing platform (EKT ePublishing at http://epublishing.ekt.gr), aiming to provide a single open access entry point to the content of scholarly eJournals, eBooks and eProceedings which have been produced through e-publishing services offered by EKT. Currently, the EKT ePublishing platform hosts fourteen academic eJournals with more than 3,000 open access articles, 14 eBooks and 43 conference proceedings.

EKT acts as the electronic publisher of Greek academic eJournals in various thematic areas, using the Open Journal Systems (OJS at http://pkp.sfu.ca/ojs/) platform which is an open source journal management and publishing system. E-Journals are peer reviewed, and each one is served by a separate, OJS installation, tailored made to the requirements and user needs of each journal.

Along with the automated process of harvesting and publishing content from external web platforms, EKT ePublishing platform enables also specified authorized users to manually import data for eJournals, eBooks and eProceedings. Figure 1 illustrates the architecture of the EKT ePublishing platform.

The content of eJournals is periodically harvested and published in the central EKT ePublishing platform, through REST-style interfaces exposed by OJS installations. The service is triggered once per day and propagates metadata updates that have been recorded in the OJS journals to the central EKT ePublishing platform.

EKT ePublishing is a web platform built with Drupal CMS in conjunction with PostgreSQL database, nginx web server and the local file system. The central platform harvests metadata from the OJS instances via REST interfaces and, in addition to that, receives data directly from authorized users to manually import content directly into the central system. A central Solr instance is used as a search engine indexing, while Apache Tika extracts the textual content from PDF files in the individual journals. As for the authentication and authorization processes, they are addressed by an LDAP Server populated with data about users organized in groups.

2.2. EKT repositories

EKT operates more than 10 open access repositories with more than 80,000 metadata records, including the Hellenic National Archive of PhD theses, the institutional repository of the National Hellenic

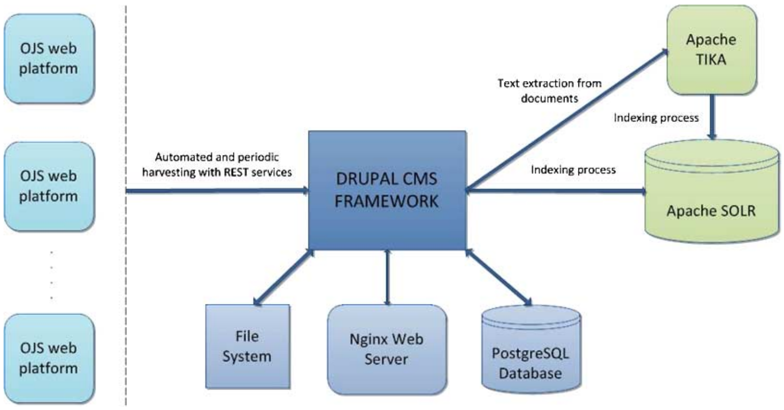

Fig. 1. EKT ePublishing platform architecture. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/ISU-140750.)

Research Foundation, a funder repository storing output of projects supported by the Greek Ministry of Education and a range of cultural repositories. The operating repositories use the shared infrastructure presented in this article for online document viewing and full text search.

3. Online reader: Infrastructures, services and tools

3.1. The rationale for an online page-by-page reader service

Currently, journal articles, books, monographs and other scientific electronic resources can be published and viewed online in various formats, such as PDF [3], HTML, EPUB, ODT. Traditionally, the most common and used format in the majority of scientific publications is PDF, because of the preserved printed format and the property protection capability, although there is an effort towards adopting the new version of EPUB format (EPUB version 3 at http://idpf.org/epub/30). Furthermore, electronic readers can be implemented in order to provide better user experience with page-by-page document reading and other significant advantages such as quicker browser loading times, abstaining from downloading large PDF files. National Documentation Centre (EKT) has integrated in all electronic publishing systems an eReader, offering greater reading capabilities. The EKT eReader is developed with the open source software Internet Archive BookReader (https://openlibrary.org/dev/docs/bookreader) and can provide a more attractive reading experience with high quality content presentation in various views (single page view, two page view, thumbnail view), allowing zoom, full text-search with hit highlighting and integration with a variety of image servers.

Certain important advantages of page-by-page online readers compared with offline reading of PDF files and embedded PDF readers (e.g. in browsers) are the following:

(a) Huge files (e.g. files well over 50 MB are commonplace in the systems described in Section 2) can be opened and navigated very quickly (low response time) and without overloading the end user's computer (low computer memory consumption), since the document is streamed on demand (one page at a time), not downloaded in its entirety and therefore opens instantly after a user click.
(b) Document opens for reading with one click (no need to download). Notably, embedded PDF readers of modern browsers have this feature as well, however this is commonly not available for

files of big size (e.g. over 50 MB), which instead get automatically downloaded by browsers or cause crashes.

(c) Each page has its own URL, which enables bookmarking and sharing at the page level, a feature particularly important in large texts such as books (although useful also for shorter documents like journal articles).

(d) Easy and fast visual navigation within the document using a grid-shaped thumbnail view screen.

On the other hand, embedded PDF readers have the advantage of copy-and-paste functionality (even for image PDFs with OCR'ed text embedded in the document), they allow easy bulk printing at the level of the entire document and of course they do not require the conversion from PDF to JPEG2000 images which is a process that needs to be executed by sophisticated infrastructure (see Sections 3.3.1 and 3.3.2) and takes time and resources.

The rest of this section elaborates on the aforementioned point (a) regarding the response time of opening a document with online reader in comparison with the same function of an embedded PDF reader or downloading a PDF file.

Table 1 includes the results of a performance test comparing the EKT online page-by-page reader "open document" time to PDF file download time, for ten levels of sizes of digital files. The sizes shown are the ones of the PDF file. The load times for each case are the median times obtained over 50 measurements on every size level. The Firebug network component has been used to perform the measurements. The tests have been performed in a client network and computing environment dedicated to these experiments, i.e. with no considerable activity from other processes of network communication. The nominal network capacity for the experiments was quite high, at the level of approximately 6 MB/s, common in high-speed academic networks and contemporary high-end residential connections.

The results show that the load time for a document in the online page-by-page reader is practically constant and independent of the size of the document – actually is around the 1 second range, which is optimal in terms of response time in web applications. This is expected, since opening a document in the reader involves rendering the first pair of pages and processing some metadata about the entire document (e.g. aspect ratios, file names for image files), which however is performed very fast and does not considerably influence response time. On the other hand, opening or downloading a PDF file requires transferring the entire file to the user's computer and is directly dependent on the size of the file.

Table 1

Load times of documents (page-by-page reader vs PDF reader)

File size (MB)	Load time page-by-page reader (s)	Load time PDF reader (s)
5	0.878	0.2
10	0.988	0.7
15	1.03	1.3
20	0.868	2
25	0.821	2.5
35	0.732	5.2
50	1.04	7.66
120	1.32	19
170	0.72	31
300	1.59	63
600	1.92	118

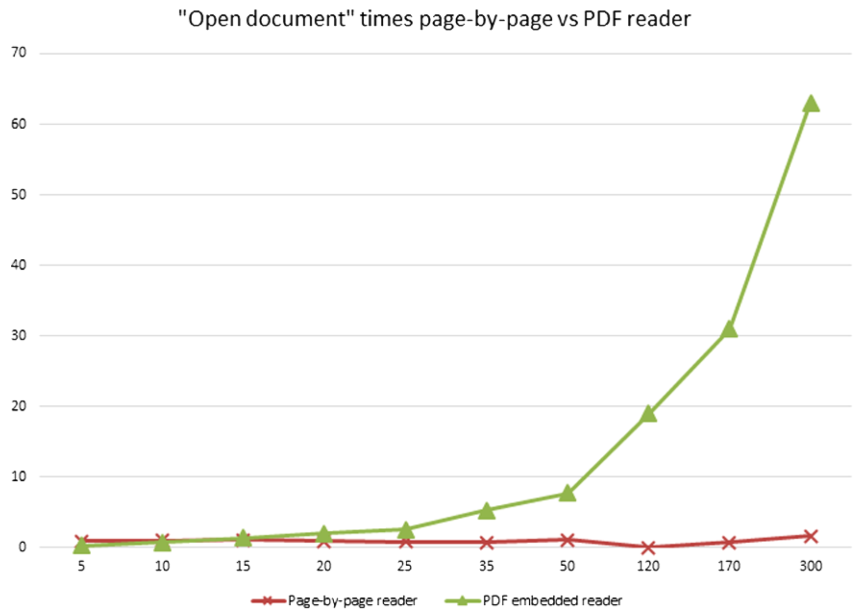

Fig. 2. Graph of load times of documents (page-by-page reader vs PDF reader). (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/ISU-140750.)

As also shown in Fig. 2, the performance advantage of the online page-by-page reader becomes considerable for PDF file sizes of 20 MB or more. The quality of user experience with embedded PDF file readers or PDF file downloads starts to become quite low with respect to response time from file sizes of 50 MB or more.

It is notable that an analysis of the file sizes in existing EKT systems confirm the need for an alternative presentation approach for large files. For example, in the entire EKT e-publishing infrastructure, out of 59 digital files of type books and conference proceedings, 44 are above the threshold of 20 MB. In the entire e-publishing infrastructure, dominated by short-sized articles, 201 digital files (around 6.5%) of the total 3069 are of size over 20 MB, still a considerable percentage. Furthermore, in a particular repository of EKT, the repository of a major funder in Greece, the Managing Authority Operational Programme "Education and Lifelong Learning",[1] which is quite heterogeneous in terms of content (e.g. books, conference proceedings, articles, educational material), out of 1767 digital files about 13.70% of the digital files are of size over 20 MB.

3.2. System architecture

EKT eReader presents the content of a PDF document page-by-page as a set of high resolution images, which are served via DJatoka image server. The process of converting a PDF file into JPEG2000 images is a fully automated process with specific workflow and requirements and can be triggered at will, from remote electronic publishing platforms as a service. The fact that the content presented in eReader is in an image format, makes full-text search process more difficult. However, EKT eReader, in its current release, supports full-text search with hit highlighting of search results. In order for the eReader to support full-text search functionality, an OCR (Optical Character Recog-

[1] http://repository.edulll.gr.

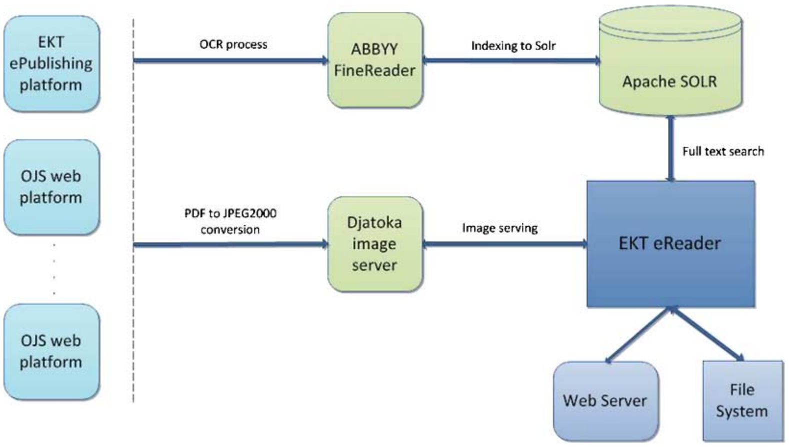

Fig. 3. EKT eReader architecture. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/ISU-140750.)

nition) software for text extraction is required as well as a search platform for content indexing. Currently, ABBYY FineReader (http://finereader.abbyy.com/) is used for the OCR process and Apache Solr (http://lucene.apache.org/solr/) as a search server, enabling among others full-text indexing and search, hit highlighting, faceted search. Figure 3 illustrates the architecture of the EKT eReader infrastructure.

3.3. *Infrastructure and services for automatic preparation of material for the online reader*

3.3.1. *Back-end image server infrastructure architecture*

While advances in PDF viewers have been made, with some of them having an increased level of interactivity and the capability to support on line reading (e.g. Multivio, https://www.multivio.org/) [2], JP2000 based viewer and JP2000 image servers still offered unrivaled browsing responsiveness, very fast browsing times and dynamic thumbnails generation among others. Since a significant amount of content is produced as “born digital”, a conversion from text-PDF to text search enabled series is required in order to enable fast and responsive online reading. JP2000 online reading with search highlighting can be supported by a mature open software stack, comprising the IA Internet Archive Book reader and the DJatoka image server. However, a critical element is missing, namely the facility to convert born digital PDFs to a series of corresponding JP2000s. Furthermore, in order for such conversion to be suitable for a large environment, some basic requirements must be met, from the software implementing them, such as:

- Capability to convert thousands of documents, with no human involvement required
- Workflow support and seamless integration with IA book reader
- Compatibility with the DJatoka Image Server
- Support of batch mode operation and online on demand conversion
- Parallel processing (in one server) and desirably distributed (over a cluster of servers) conversion

In order to provide such page by page viewing experience the following back-end PDF conversion and image delivery components were integrated with the epublishing platform:

- A conversion platform comprising a multithreaded distributed conversion system, in order to interface with the epublishing platform and manage the batch conversion process from PDFs to JP2000 files. The conversion platform is required to be decoupled for the delivery layer. EKT has developed

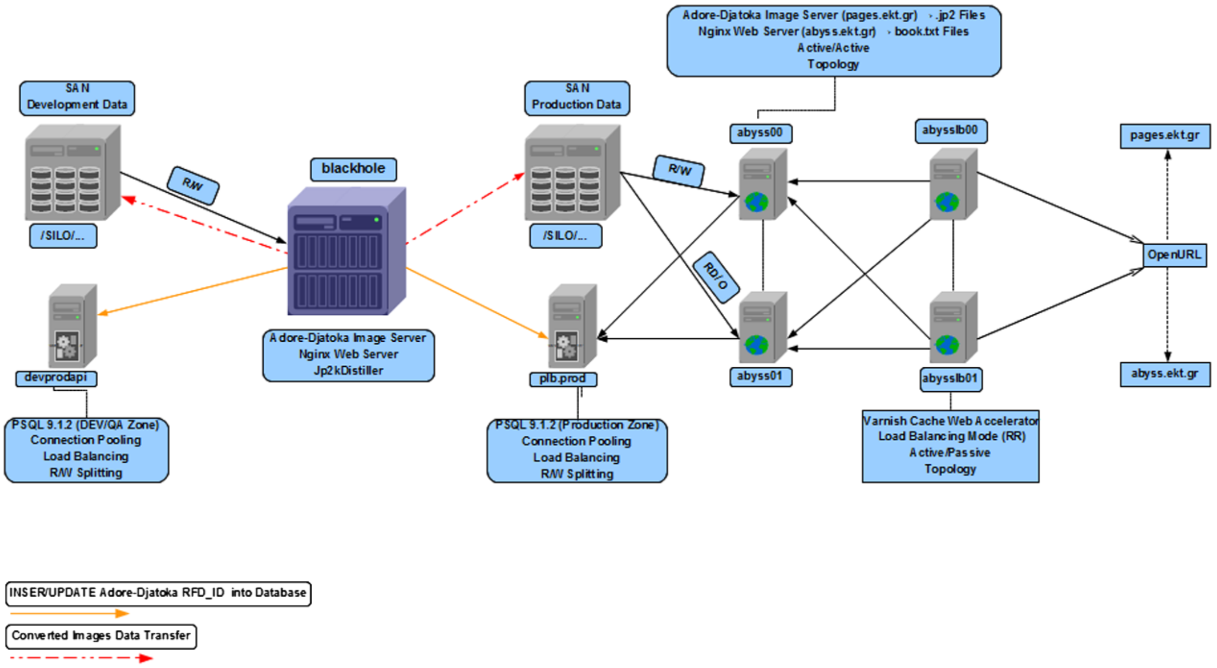

Fig. 4. EKT eReader architecture. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/ISU-140750.)

the JP2K-Distiller, which is Python based set of scripts that manage the deployment, scheduling and distribution of conversion jobs, and the interfacing with the external components, i.e. the epublishing platform, OJS eJournals and DSpace repositories. JP2K-Distiller is highly distributable, in its second version, and can scale to an arbitrary number of processing nodes. File conversion is broken into several batch processes according to the active processing nodes and the conversion process is automatically delegated to the processing nodes. Furthermore functionality for the monitoring of the conversion process and log messages is provided.

- A highly scalable image delivery layer. The DJatoka image server is utilised, over a clustered installation forming a shared clustered deployment, which exploit multiple virtual servers over a shared storage, with advanced load balancing and failover capabilities.

This two-tier platform (delivery and conversion tiers), depicted in Fig. 4, comprises a large scale image conversion and delivery platform, which is capable of driving large scale digital content systems and is exploited as a common service element among the publishing platform and other digital content systems (repositories and eJournals). The platform is implemented in the system level of a Nginx/Varnish failover load balancer/caching to 2 node tomcat/djatoka cluster serving JPEG2000 over a shared SAN storage and utilizes a 3 node Postgres 9 cluster.

3.3.2. *Integration with external systems* (*OJS*, *Drupal*, *DSpace*)

EKT has integrated online reading capability in several platforms, such as the central ePublishing platform (http://epublishing.ekt.gr/) which is developed with Drupal, eJournals management systems developed with OJS (e.g. http://www.medit-mar-sc.net/) and digital repositories (e.g. http://repository.edulll.gr/) developed with DSpace. The EKT eReader is provided as a common service shared across platforms and interoperates with them using a loose coupling approach.

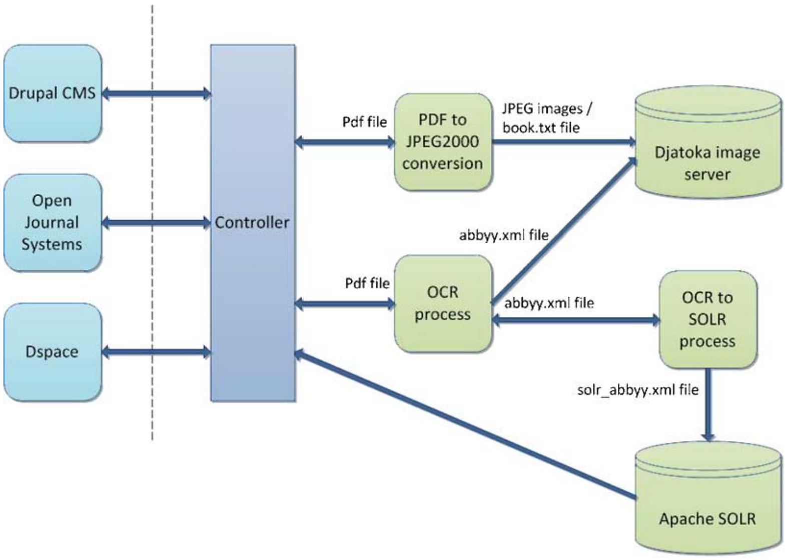

Fig. 5. Integration with external systems. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/ISU-140750.)

An important part of this interoperation is the workflow which is executed when new material (e.g. PDF files) is uploaded in any of the aforementioned platforms. The result of the workflow is that the generation and appropriate placement in the system of the entire set of artifacts (i.e. JPEG 2000 images, OCR results, indexing results) needed to publish the new material through the online reader. Furthermore, the links to the documents in the reader are made available to end users within the respective platform (Drupal, OJS or DSpace).

This workflow is depicted in Fig. 5 and can be described as follows: Each platform triggers the process via a REST request to a central "controller" which receives specific data for the new material, such as document title and authors, PDF file URL and the corresponding publishing platform (Drupal, OJS or DSpace). The controller triggers the process of converting PDF files to JPEG2000 images using the solution described in Section 3.3.1. Images are transferred to the Djatoka image server with a specified structure (e.g. file paths, identifiers) for each document and publishing platform, accompanied with a relevant text file describing this structure (book.txt file). The information in the text file includes data such as the number of images/pages per document, list of file names (reflecting order of images/pages in the document). Once the conversion process is successfully finished, the controller is notified by the conversion systems and informs the corresponding publishing platform through a callback invocation which includes as a parameter the URL of the document in the online reader. After successful conversion, the process to provide full-text search for the new document(s) is triggered, with Optical Character Recognition (OCR) as the first step; a commercial software server (ABBYY) is used for that purpose. In case the output file (abbyy.xml file) from the OCR process meets minimum quality requirements (i.e. the text is recognizable by OCR), the file is further processed by a python script (solr_abbyy.xml file) so that it can be indexed in Apache Solr. When the indexing process is finished, the controller informs the corresponding platform in order to enable full-text search in the online reader. The whole process is triggered every time a new pdf file is uploaded in the publishing platforms for specified content types. The described EKT eReader extension has been implemented in Drupal, Open Journal Systems and

DSpace and can be extended in order to meet the requirements of further systems, since the controller hides much of the complexity of the workflow implementation.

3.4. EKT eReader – Additional features

3.4.1. Addressing diversity of images sizes and aspect ratios

A page by page reader presenting digital content in an image format, has to maintain uniformity of image aspect ratio and it is particularly important that image dimensions are determined by the current screen size, enabling better reading experience in mobile devices too. Moreover, the images in a document are not always in the same dimensions, considering that they might come from scanning process and be further processed. EKT eReader adopts an image processing algorithm in order to present all images in the same dimensions, by using each image width and height information provided by Djatoka image server. Image dimensions are determined according to the computation of the average aspect ratio for each document and change dynamically when screen size is altered, showing always the image with the appropriate resolution.

Furthermore, in many cases, there are documents which consist of pages with both portrait and landscape orientation, illustrating mostly large tables and wide images. Usually, a page-by-page reader implements the appropriate algorithms in order to publish and present images with standard dimensions in portrait orientation, resulting in the distortion of landscape pages. Other readers avoid using an image processing algorithm, thus, enable portrait and landscape pages to be published in their real dimensions. However, EKT eReader uses an algorithm that recognizes if there is a page in landscape orientation within a document. If a document contains at least one landscape page, then all the landscape pages are rotated 90 degrees (in two-page view) so that they are presented as portrait, avoiding images distortion and retaining the same dimensions for all pages. Furthermore, a relevant button appears next to the rotated landscape page, allowing to view the page in its real dimensions by clicking on it. Implementing this algorithm, consistency is preserved as regards to the size of pages, without distortion and quality reduction. Figure 6(a) illustrates a document with both portrait and landscape pages embedded in EKT eReader.

3.4.2. Bookmarking at the page level

EKT eReader is also customized in order to provide the ability to create a distinct URL not only for the entire document, but also for each page, allowing bookmarking and sharing on popular social media,

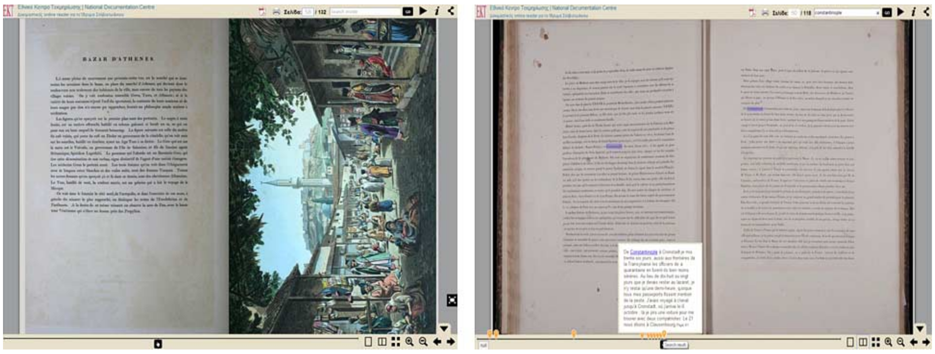
(a) (b)

Fig. 6. (a) EKT eReader with both portrait and landscape pages. (b) EKT eReader search results with hit highlighting. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/ISU-140750.)

at page level. A unique OpenURL-compatible URL is available for every page, both at the JP2000 image level and at the reader page level. This capability is provided by the features of the Djatoka server where a unique identifier exists for each image (page). The format of the address is configurable through the Book Reader.

3.4.3. Printing the page level

EKT eReader enables the user to print each page in different image resolution, based on the current page zoom level. This is accomplished with the ability of Djatoka image server to determine multiple resolutions for each image, thus, printing a page in EKT eReader depends on the selected page zoom level, which corresponds to a specific image resolution in Djatoka.

4. Infrastructure and services for full-text search

4.1. Full-text search integrated into the online reader

Full-text search service requires initially the initiation of the OCR process for a specific PDF file and the creation of the output file in XML format, which contains the coordinates of every single character, word, row or paragraph in the whole page. This process is critical for the success of full-text search service, thus, the quality of the output file must meet the appropriate requirements so that the process can proceed.

Thereafter, the output XML file is properly processed in order to be indexed in Solr search platform. Once the processed XML file is created and transferred into the Solr instance, Apache Solr is triggered to start the indexing process for all the new files that exist in Solr instance. EKT eReader implements full-text search functionality if the above actions are successfully completed, enabling the user to search inside the document. EKT eReader search process executes a query in Solr search platform and the returning results are matched with the initial XML file in order to provide targeted highlighting of search results in a quite interactive way. Figure 6(b) illustrates the search results for a specified keyword with hit highlighting in EKT eReader.

4.2. Distributed full-text search in the EKT ePublishing platform

The content of EKT ePublishing platform is constantly growing in a way that the integration with a search platform for distributed indexing is essential, in order to quickly retrieve specific information. Furthermore, EKT ePublishing platform provides access to the full text of open access electronic publications which is in pdf format, in addition to the rest available metadata. Currently, one of the most popular and robust search platforms is Solr (http://lucene.apache.org/solr/) from the Apache Lucene project. Solr is an open source standalone search server which can be used as a replacement for traditional content search, providing features such as faceted search, hit highlighting, full-text search, near real-time indexing and dynamic clustering, and boosting the performance of a web application. Integrating Solr with EKT ePublishing is a common process, which leads to Drupal's core content search replacement and the implementation of all the extra features of Solr. Indexing process of EKT ePublishing platform with Solr is triggered once each day, keeping the indexes up to date when new content is added in the platform or existing content is modified or removed.

Moreover, Apache Tika (http://tika.apache.org/) has been integrated in Solr search platform, for document text extraction, in order to provide full text search functionality. Tika is a project of the Apache Software Foundation (http://www.apache.org/) for metadata and text content extraction from various documents and can be easily implemented with Solr and Drupal CMS. EKT ePublishing platform allows electronic publications' pdf files to be uploaded in the corresponding installation file system. As for the electronic publications that are imported in the platform from external electronic journals systems, the source files of the publications are not transferred to the EKT ePublishing platform's file system, for better performance and statistics management. As a result, not only the local files that are manually uploaded to the platform, but also remote pdf files from various external web platforms, are indexed in the same search platform, enabling distributed full text search in the EKT ePublishing platform with hit highlighting and faceted search, among other features. Implementing Solr provides also more advanced search capabilities and makes it possible to create relevant content blocks for each page showing similar content based on specific attributes. EKT ePublishing platform[2] supports search functionality with hit highlighting, facet filters and autocomplete search box.

5. Summary

This work concerns a set of infrastructural components and their utilization for the creation of related coherent services and features for end users. We present a set of sophisticated platforms, tools and mechanisms that have been employed in real-life cases for implementing document viewing and full-text search features, shared among application instances. Challenges encountered and the provided solutions are discussed.

Acknowledgements

The work presented in this article has been partly supported by the projects "National Information System for Research & Technology/Social Networks-User Generated Content" (Ref. No 296115) and "Platform for provision of services for deposit, management and dissemination of Open Public Data and Digital Content" (Ref. No 327378). Both projects are co-funded by Greece and the European Union-European Regional Development Fund through the Operational Programme "Digital Convergence" (NSFR).

References

[1] E. Tzoc, Document viewers for non-born-digital files in DSpace, *Journal of Digital Information* **13**(1) (2012).

[2] M. Moreira, Multivio, a flexible solution for in-browser access to digital content, in: *7th International Conference on Open Repositories*, Edinburgh, 2012.

[3] Y. Zhou, Are your digital documents Web friendly?: Making scanned documents Web accessible, *Information Technology and Libraries* **29**(3) (2010), 151–160.

[2] http://epublishing.ekt.gr/.