Quality assured image file format migration in large digital object repositories

Using various outcomes of the SCAPE project in the context of library preservation scenarios

Sven Schlarb
Austrian National Library
sven.schlarb@onb.ac.at

Peter Cliff, Peter May, William Palmer
British Library
{peter.cliff, peter.may, william.palmer}@bl.uk

Matthias Hahn
FIZ Karlsruhe
matthias.hahn@fiz-karlsruhe.de

Reinhold Huber-Moerk, Alexander Schindler, Rainer Schmidt
Austrian Institute of Technology GmbH
{reinhold.huber-moerk, alexander.schindler, rainer.schmidt}@ait.ac.at

Johan van der Knijff
National Library of the Netherlands
johan.vanderknijff@kb.nl

ABSTRACT

This article gives an overview on how different components developed by the SCAPE project are intended to be used in composite file format migration workflows; it will explain how the SCAPE platform can be employed to make sure that the workflows can be used to migrate very large image collections and in which way the integration with a digital object repository is intended.

Two institutional image data migration scenarios are used to describe how the composite workflows could be applied in production library environments. The first one is related to the British Newspapers 1620-1900 project at the British Library which produced around 2 million images of newspaper pages in TIFF format. The second is a large digital book collection hosted by the Austrian National Library where the book page images are stored as JPEG2000 image files.

1. INTRODUCTION

Several memory institutions in the SCAPE project, such as the British Library, the National Library of the Netherlands, and the National Library of Austria are using the JPEG2000 image file format for storing images of digital newspapers, books, or other image collections.

Due to advantages of the JPEG2000 file format, like the ability to reduce storage costs by using lossless and lossy compression, many institutions have migrated (or are planning to migrate) their TIFF master images. There is, therefore, a clear need for systems capable of transforming huge amounts of images into the new format and for making sure that no information is lost during this process. While this article will focus on migration scenarios from TIFF to JPEG2000 and JPEG2000 to TIFF, the topic is actually more generic; it is about image file format migration of large collections and the question of how this preservation action is embedded in more complex production-ready data migration workflows.

In this context the SCAPE project (SCAlable Preservation Environments), partly funded by the European Commission, is doing research and providing solutions that help memory institutions in performing preservation at scale. The project develops an execution platform together with preservation tools and advanced services for preservation planning and watch. Development is driven by institutional requirements and tested in real world institutional environments in order to ensure that the solutions are really applicable on diverse data sets and on a large scale.

This article will give an overview in which ways different components developed by the SCAPE project are intended to be used in composite file format migration workflows, explaining how the SCAPE platform makes these workflows scalable so they can be used to migrate very large image collections. Furthermore, it will discuss the implications that the use of the SCAPE platform has on development and integration of the different components.

We start by explaining the institutional image migration scenario in more detail. We then outline the SCAPE components used in the composite workflows, before presenting the composite workflows themselves. Finally, we conclude the article with a summary and outlook.

2. THE INSTITUTIONAL SCENARIOS

Our first scenario is a real world use case of the British Newspapers 1620-1900 project at the British Library which was funded by the Joint Information Systems Committee (JISC) and produced around 2 million images of newspaper pages in TIFF format [1]. In order to reduce the storage cost of these images, the British Library undertook a migration of the items to the JPEG2000 format prior to ingest into the Digital Library System.

Our second scenario looks at a large digital book collection hosted by the Austrian National Library where the book page images are stored as JPEG2000 image files and serve as master and access copies at the same time. First, there is the requirement to harmonise file formats being used in different collections which is the reason why the migration of legacy TIFF based image collections to the JPEG2000 format is being considered. Additionally, large scale migration workflows must be available in case the JPEG2000 profile is to be changed or a decision is made to go back to the TIFF image format.

For both of these scenarios it is clear that a system capable of migrating millions of images from one format to another is required. Workflows executed on this system must include steps that validate both original and migrated image files and provide assurance that migration was successful and produced equivalent migrated images and valid instances of the new format. Typically both original and migrated formats will be stored in digital repositories and so we must also consider both access to the original and ingest procedures to store the results.

Generally, we are claiming that a robust large-scale migration system must be capable of detecting and reporting the following:

1. The validity of the original file.
2. The opinion of the migration tool as to its own success or failure.
3. The conformance of the migrated file to any given profile (where profiles are part of the output format).
4. The completeness of the migrated file (i.e. is all the data intact).
5. The validity of the migrated file (reporting where, if at all, it deviates from a specification - this is to enable the user to decide if lack of validity is an issue or not, c.f. PDF/A).
6. That any other requirements are met by the migration - e.g. the migrated file is smaller than the original.

In the following section we describe the SCAPE tools that are used in the composite workflow and which are essential to fulfilling the requirements listed above.

3. PRESERVATION COMPONENTS

As briefly mentioned in the introduction, the SCAPE project is developing new components, extending and improving existing tool implementations and providing means for integration of new tools into the SCAPE preservation platform.

[1]http://www.jisc.ac.uk/media/documents/programmes/digitisation/digitisation_brochure_v2_overview_final.pdf

In order to give a complete picture about how software components of different types can be put together in a composite workflow, we make use of two tools developed in the SCAPE project which will be described in more detail in the following sections.

3.1 Jpylyzer

Jpylyzer [10] is a validator tool for the JP2 (JPEG 2000 Part 1) still image format. It was developed with the following uses in mind:

- verification of whether an encoder produces standard-compliant JP2s;
- detection of JP2s that are corrupted (e.g. images that are truncated or have missing data);
- extraction of technical characteristics and metadata.

Although some of the above features are also provided by other software tools, these either provide limited or incomplete validation functionality, partial coverage of JP2's feature set, or produce output that is difficult to interpret. The main philosophy behind Jpylyzer was to create a tool that strictly adheres to the JP2 format specification, is lightweight, simple to use and scalable. The validation procedure includes a verification of the general file structure, tests on the validity of individual header fields, and a number of consistency checks.

3.2 Matchbox

The Matchbox tool was designed for content based image characterization and comparison. It is based on robust detection and invariant description of salient image regions using the Scale Invariant Feature Transform (SIFT) [7]. Categorization of image content uses the Bag of Features (BoF) approach [2] which is inspired by the bag of words approach in information retrieval. In the BoF approach scanned book pages are characterized by compact visual histograms referring to visual words contained in the BoF. The BoF itself is constructed for each collection, i.e. a book scan, using machine learning. Once the BoF is created, image comparison becomes an efficient comparison of histograms. Matchbox also implements detailed image comparison based on the estimation of a geometric transformation between pairs of images followed by the estimation of a perceptual measure of Structural Similarity (SSIM) [11].

Currently, there are three basic modes of operation for Matchbox in image quality assurance workflows:

Comparison of images the content of image pairs is compared allowing tone/colour modifications as well as geometrical differences such as different rotation, scaling and cropping. Similarity is expressed by SSIM, where 1 means perceptually identical and 0 is perceptually different.

Comparison of collections two collections of images, typically two differently acquired scans of the same collection, are analyzed in order to associate individual pages between collections and detect missing pages and, finally, assess the visual similarity of associated pages [4].

Duplicate detection within a collection the replication of individual pages within a single collection is detected and visual similarity of image content is quantified [5].

In regards to the workflows presented in section 5, only the first mode is used.

4. EXECUTION PLATFORM

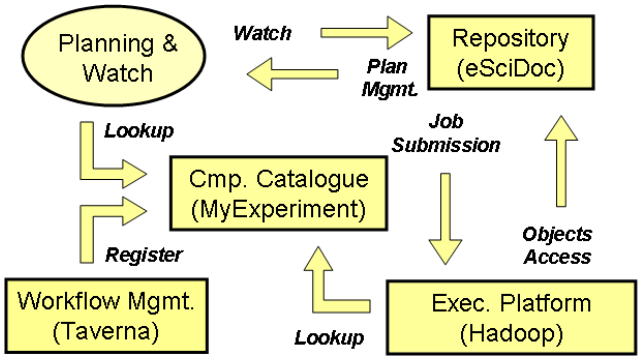

Figure 1: Components and services of the SCAPE Preservation Platform. The available software components provide support for workflow design and description, registration and lookup of preservation components, scalable storage and execution, and digital object management and efficient access. Integration with the SCAPE Preservation Planning and Watch components is supported through the Component Catalogue Lookup API and the Repository Plan Management and Watch APIs.

The SCAPE Preservation Platform [8] provides an infrastructure that targets the scalability of preservation environments in terms of computation and storage. The goal is to enhance the scalability of storage capacity and computational throughput of digital object management systems based on varying the number of computation nodes available in the system. A platform instance is based on existing, mature software components like Apache Hadoop[2], the Taverna Workflow Management Suite[3], and the Fedora Digital Asset Management System[4]. The platform implements a set of additional services on top of these software components to specifically support scalability and integration with digital preservation processes as well as to integrate with other SCAPE components, such as the SCAPE preservation watch system, SCOUT [1]. Figure 1 provides an overview of the main software components of the SCAPE preservation platform and shows their interactions.

A key challenge of the platform is the development of methodologies to integrate preservation tools with its parallel execution environment. The automated deployment of preservation tools such as Jpylyzer, described in section 3.1, is based on software packages like those maintained by the Open Planets Foundation [5] and a Linux based software package management system (presently based on Debian). Complex software environments like pre-configured platform nodes can be deployed on virtualized hardware using virtual machine images[9]. The platform provides support for migrating existing and sequential preservation workflows and applications to the parallel environment covering different aspects like data decompostition, tool handling, workflow support, or repository interaction. However, the strategy used to parallelize an individual workflow depends on the use case it implements and may be selected on a case-by-case basis. Section 4.3 discusses basic parallelization approaches with respect to the example workflow discussed in this paper. A flexible mechanism for the integration of existing digital repository systems is provided by the SCAPE Data Connector API. This generic interface supports the efficient exchange of data sets between the execution platform and digital object management systems like the SCAPE repository reference implementation, described below.

[2]http://hadoop.apache.org
[3]http://taverna.org.uk
[4]http://http://www.fedora-commons.org
[5]http://deb.openplanetsfoundation.org

4.1 Digital Object Repository

The SCAPE platform provides a Digital Object Repository to allow storage and management of digital objects. The SCAPE repository (eSciDoc[6]), based on Fedora Commons [7] is a joint project of the Max Plank Society [8] and FIZ Karlsruhe [9]. The repository offers several APIs to integrate with the SCAPE platform and other SCAPE components like Planning and Watch. Preservation actions running on the execution environment are able to interact with the repository via a RESTful service API. This Data Connector API allows ingest, retrieval, update and query of a repository's content.

A Digital Object Model has been defined to allow different SCAPE components to exchange data in a standardized way. This model is based on METS[10] as a container format, along with other metadata formats like Dublin Core[11], Marc 21[12], PREMIS[13] and other technical, administrative and rights metadata. The data we are focusing on is already provided in a METS format and as such can be ingested into the repository via a Loader Application, briefly described in the next section.

4.2 Loader Application

The SCAPE Loader Application is a Java-based client application with different input source options (local or distributed file system). Its intended use is for ingesting a large amount of digital objects (represented as METS) into the repository using the REST endpoint defined by the Data Connector API. It monitors and logs the ingest process, e.g. retrieves the life-cycle status of each digital object of the repository. Figure 2 illustrates the ingest process sequence.

4.3 Scalable Processing

The SCAPE preservation platform utilizes the Apache Hadoop framework as the underlying system for performing data-intensive computations and consequently relies on MapReduce [3] as the parallel programming model. In SCAPE, preservation scenarios are typically developed as sequential workflows using desktop tools like the Taverna workbench. Such conceptual workflows, which will be explained in more detail in section 5, define the general logic of a preservation

[6]eSciDoc, https://www.escidoc.org/JSPWiki/en/Overview
[7]http://fedora-commons.org/
[8]http://www.mpg.de
[9]http://www.fiz-karlsruhe.de
[10]http://www.loc.gov/standards/mets/
[11]http://dublincore.org/
[12]http://www.loc.gov/marc/bibliographic/
[13]http://www.loc.gov/standards/premis/

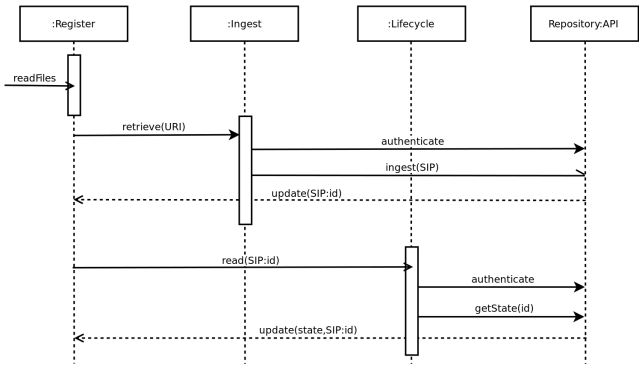

Figure 2: scenario diagram of the SCAPE Loader Application.

scenario and must be migrated to the parallel environment before they can be executed on the SCAPE preservation platform at scale.

Depending on their complexity, preservation workflows (or activities within a workflow) can be turned automatically into a parallel application that runs on the platform to a certain degree. An example is the execution of preservation tools against large volumes of files which can be performed on the platform using a generic MapReduce tool wrapper. The SCAPE tool specification language supports users in selecting a particular tool and parameter configuration used during the execution. SCAPE has also developed a model allowing a workflow designer to describe preservation activities following a defined component specification and register them to the SCAPE Component Catalogue (c.f. figure 1). The platform makes use of this approach to discover runtime dependencies of workflows, like dependencies on preinstalled software packages, which must be resolved prior to workflow execution.

However, as discussed in this paper, it is typically required to migrate more complex workflows involving different activities, data flows, and decision logic to the platform environment. A simplistic approach is to instantiate and concurrently execute multiple instances of the sequential workflow on a range of cluster nodes. This strategy however comes with a number of restrictions as compared to an approach where the workflow language is fully translated into a native MapReduce program, a strategy which is also evaluated in the context of SCAPE.

5. WORKFLOWS

As already mentioned, Taverna [6] is used in the SCAPE project to build composite workflows using the components described in section 3.

5.1 Workflow development and testing

Figure 3 shows the diagram of a Taverna workflow that has been designed for single-threaded execution in order to get experimental results for the execution of the workflow as a whole as well as for the different components. The diagram has an input "inputUrl" at the top and the workflow output "report" at the bottom. The input port takes, as input, a local (file://) or remote URL (http://) pointing to a text file which itself contains local file paths to the image files. Highlighting the core components of the workflow, "jpylyzerValidation" is based on the Jpylyzer tool that validates the JPEG2000 input file, "j2k_to_image" decodes the JPEG2000 image using the JPEG2000 image library Kakadu[14], "Jhove2Validation" uses JHove2[15] to validate the output TIFF file, and finally "MatchboxComparison" based on the Matchbox tool performs a feature comparison of original and migrated images in order to verify if the migration was successful.

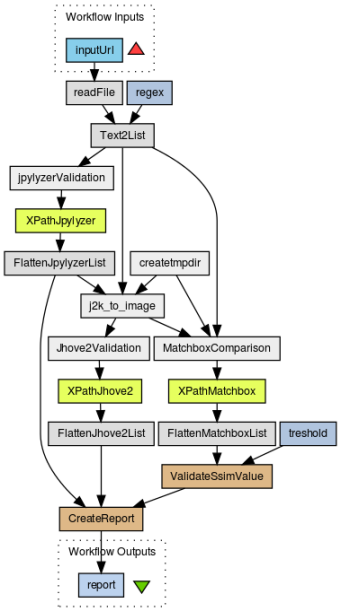

Figure 3: Experimental non-distributed workflow; http://www.myexperiment.org/workflows/3399

By processing a representative sub-set of images as a sequence of single-threaded processes, experiences can be extrapolated to the entire collection that is going to be migrated. First of all, the results of the execution give insight into the viability and robustness in terms of possible errors using a controlled, non-distributed setting. Furthermore, with a large-scale migration in mind, results such as the execution time, memory usage, and size of any intermediate or final output files all provide important information to estimate the overall hardware requirements for the migration of the larger collection, as well as inform scalable workflow design.

5.2 Example large scale workflow

The workflow in Figure 4 shows the steps required to migrate a TIFF to a JP2 and quality assure the results. It was designed to address the requirements of the British Library's TIFF to JP2 migration scenario. Input to this workflow is a list of TIFF files and the output is the migrated JP2s and a report giving details of the migration and quality assurance stages. The workflow consists of both sequential and parallel layers. For example, once the TIFF to JP2 migration completes (HadoopMigrate) then metadata extraction, feature

[14]Version 6.3.1, http://www.kakadusoftware.com
[15]https://bitbucket.org/jhove2/main/wiki/Home

extraction using Matchbox and profile validation using Jpylyzer can all operate on that JP2 at the same time. Similarly, while TIFF to JP2 migration is taking place, the workflow can also be extracting features from the TIFFs using Matchbox ready for comparison with the features extracted later from the JP2.

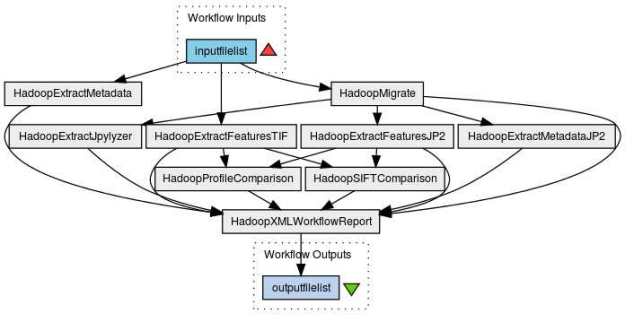

Figure 4: SCAPE Platform migration TIFF to JP2; http://www.myexperiment.org/workflows/3400

5.3 Workflow implementation methods

Having created a single-threaded sequential Taverna workflow, as noted in the platform section 4, it is sometimes necessary to translate this into a suitable MapReduce program for execution on the SCAPE Platform. Performing actions like file migration using Hadoop is achieved by using one or more map jobs (made up of many map tasks) across a number of processing machines and few (if any) reduce jobs. When translating a workflow like this we need to decide what each map task should do. We have explored several ways to do this.

5.3.1 Vertically aligned workflow

Orienting the workflow with inputs at the top and output at the bottom, one option is to slice the problem vertically and execute the workflow from top to bottom for every input file. Here each map task calls the Taverna command line with a single input file and the workflow definition. Taverna is responsible for the order of execution within the workflow and runs steps in parallel, where possible, according to the workflow graph.

This vertical slicing has a number of advantages. Taverna preservation workflows that work well on single machines can easily be scaled using the SCAPE Platform. Workflow designers do not need knowledge of Hadoop and workflows can be re-used. This is the idea behind SCAPE components. We can also make use of Hadoop's robust design: should the workflow fail, that map task fails; Hadoop will handle retrying the map task and reporting the failure. Many workflows will create intermediate files on the processing data node. Doing all the work on a single data node avoids moving these files across the Hadoop cluster and managing their locations. Finally, Hadoop requires no knowledge of Taverna, and (unless using HDFS) the workflow does not need any knowledge of Hadoop.

Of course, this simplicity comes at some cost. Each map task starts its own instance of Taverna which adds startup cost and memory overhead to the processing. If each map task executes a workflow that includes parallel execution this begs the question: would it be better to re-write the workflow to take into account this parallel execution and enable Hadoop to manage these parallel tasks?

5.3.2 Horizontally aligned workflow

Another option is to slice the problem horizontally and execute each layer of the workflow as a chain of map tasks[16]. For the workflow presented in Figure 4 the TIFF to JP2 migration is performed over all files, one map task per migration. At the same time a second set of map tasks can be extracting the features and metadata of the TIFFs. Once complete another set of map tasks extract features from the JP2s and so on. It is clear that something is needed to manage this execution and for this we can use Taverna. However, this approach requires that the sequential workflow be re-written with knowledge of Hadoop. This only requires a single instance of Taverna, but, perhaps surprisingly, current small scale tests indicate that there is no additional performance gain executing the workflow in this way[17]. A significant disadvantage of this approach is that currently there is little integration between Hadoop and Taverna. Taverna cannot, for example, report on the progress of the map tasks. It is also difficult to resume (rather than restart) the workflow at the appropriate point in the dataset in the event of error.

5.3.3 Translation to MapReduce

A final option would be to translate the Taverna workflow to one or more native Hadoop jobs, using Taverna to design the workflow but not using it during execution. This strips away a layer of complexity and offers the most raw performance gain, but also requires an experienced MapReduce developer to do the translation.

5.3.4 Overall thoughts

It should be clear from these discussions that no one approach to executing Taverna-designed digital preservation workflows on Hadoop provides the perfect solution. The choice of which to use depends on the needs of the institution and the skill set available. A slow running, sequential workflow would work well for a project where Java programmers are unavailable and execution time is less important, but for maximum throughput a knowledge of the scalable platform and a willingness to redesign the workflow is required. There are many unanswered questions and the SCAPE project will continue to investigate best practice in this area.

5.4 Storage and retrieval of files

Another consideration when using Hadoop is where to store the data to process. For these workflow experiements all the data, including intermediate and final results, were stored in Hadoop's own file system, HDFS. Hadoop uses this to try to guarantee data locality - that the data being processed is being stored on the node doing the processing and in general this helps keep Hadoop performance high. However, the integration of the migration workflows with a digital object repository, like the one described in 4.1 brings the scenario closer to the real world institutional environment

[16] http://openplanetsfoundation.org/blogs/2012-08-07-big-data-processing-chaining-hadoop-jobs-using-taverna

[17] http://www.openplanetsfoundation.org/blogs/2013-02-14-mixing-hadoop-and-taverna

of libraries and archives, where the data is not necessarily local to the processing. It is therefore important to compare performance indicators, like the throughput in terms of processed items per unit of time, including the time needed to retrieve input data from the repository and loading the migrated objects back into the repository. Furthermore, any additional data preparation, like loading the data into the distributed storage system or transforming data into a format that is suitable for processing in the SCAPE Platform, must be taken into account. In addition, these factors can also influence workflow design. For example, where the migration process has a long execution time, we should be able to ignore disk access and network times retrieving the originals from a repository. Where the migration process completes more quickly we may prefer to design a workflow that starts by moving the data as a batch to the processing platform. We intend to do more work to explore the impact of these factors on scalable preservation workflow design.

5.5 Digital objects repository integration

The JPEG2000 to TIFF migration scenario using the digital book collection of the Austrian National Library provides a production environment for testing the large scale applicability and for gathering performance indicators related to the approach. In this context, a digital book object is defined by a METS container which, among other things, aggregates the digital book page entities (each page consisting of an image, full text, and full HTML layout representation) and contains references to the physical files on the file server. This METS container is the input for the Loader Application described in section 4.2. According to the current setup of the test environment with access to the complete set of production data, image data remains on the file server and is not loaded into the distributed file system (nor the repository) because this would exceed the storage that is available on the Hadoop cluster. All the other files (METS container file, full text, and full HTML layout representation) are bundled in large Hadoop SequenceFile[18] input format files and stored in the distributed storage. The workflow that integrates with the digital object repository first ingests the METS container files - the submission information packages (SIPs) according to the OAIS reference model - into the repository. The binary content (images etc.) will only be referenced by the repository. Then the workflow is executed and the migrated images are added as new representations to the intellectual entity (the digital book object). Towards the end of the SCAPE project, an evaluation will be made of overall system and component level performance indicators.

6. CONCLUSIONS

In this article we have presented several core outcomes of the SCAPE project along with preservation scenarios that give a better idea of how they can be used in an institutional context. We have also shown how tools can be used in workflows combining characterisation, migration, and quality assurance tasks.

According to the SCAPE project's mission to provide solutions that work on a large scale, we have discussed approaches to transform conceptual workflows into workflows which can be executed on the SCAPE platform and integrated with a digital object repository.

[18]http://wiki.apache.org/hadoop/SequenceFile

The development of these workflows will be pursued further this year; towards the end of the project, evaluations will give more insight into performance, runtime stability and organisational fit of the solutions presented in this article.

7. ACKNOWLEDGMENTS

This work was partially supported by the SCAPE project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

8. REFERENCES

[1] C. Becker, K. Duretec, P. Petrov, L. Faria, M. Ferreira, and J. C. Ramalho. Preservation watch: What to monitor and how. In *Proc. of the Ninth Int. Conf. on Preservation of Digital Objects (iPres12)*, Toronto, Canada, October 2012.

[2] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV 2004*, pages 1–22, 2004.

[3] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51:107–113, January 2008.

[4] R. Huber-Mörk and A. Schindler. Quality assurance for document image collections in digital preservation. In *Proc. of Advanced Concepts for Intelligent Vision Systems ACIVS 2012*, volume 7517 of *Springer LNCS*, pages 108–119, Brno, CZ, Sep 2012.

[5] R. Huber-Mörk, A. Schindler, and S. Schlarb. Duplicate detection for quality assurance of document image collections. In *Proc. of Conf. on Digital Preservation iPres 2012*, pages 136–143, Toronto, CA, Oct 2012.

[6] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn. Taverna: a tool for building and running workflows of services. *Nucleic Acids Res*, pages 729–732, 2006.

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Comput. Vision*, 60(2):91–110, 2004.

[8] R. Schmidt. An architectural overview of the scape preservation platform. In *Proc. of the Ninth Int. Conf. on Preservation of Digital Objects (iPres12)*, Toronto, Canada, October 2012.

[9] R. Schmidt, D. Tarrant, R. Castro, M. Ferraira, and H. Silva. Guidelines for deploying preservation tools and environments. Technical report, SCAPE Project Deliverable, March 2012.

[10] D. Tarrant and J. V. D. Knijff. Jpylyzer: Analysing jp2000 files with a community supported tool. October 2012.

[11] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600 –612, April 2004.