Improving Access to Text
IMPACT

D-OC5.3

Evaluation Report | OC5

External

Revisions

Version	Status	Author	Date	Changes
0.1	Draft	Clemens Neudecker, Mustafa Dogan, Sven Schlarb	3 June 2010	Created
0.2	Draft	Clemens Neudecker	26 July 2010	Input from OC5/TR/EE included
0.3	Pre-final	Clemens Neudecker	5 August 2010	Initial comments processed
1.0	Final	Clemens Neudecker	30 August 2010	Internal review comments processed

Internal Review

Version	Status	Reviewer	Date	Role in project	Approval
0.3		Neil Fitzgerald	9 August 2010	WP Leader OC2/OC4	OK
0.3		Günter Mühlberger	10 August 2010	SP Leader TR	OK

Approvals

This document requires the following approvals:

Version	Date of approval	Name	Role in project	Signature
1.0	30 August 2010	Aly Conteh	SP Leader OC/CB	OK
1.0	30 August 2010	Hildelies Balk	General Project Manager	OK

Distribution

This document was sent to:

Version	Date of sending	Name	Role in project
0.1	9 July 2010		OC5 WP members
0.2	26 July 2010		OC5 WP members
0.3	5 August 2010		Internal reviewers
1.0	1 September 2010		European Commission
1.0	1 September 2010		All staff (sharepoint)

KB IMPACT is supported by the European Community under the FP7 ICT Work Programme. The project is coordinated by the National Library of the Netherlands

Table of contents

1. Executive summary

A core piece of work in IMPACT lies in the development of novel software techniques for a number of tasks connected to Optical Character Recognition (OCR), such as image enhancement, segmentation and post-processing, as well as in the improvement of existing OCR engines and experimental prototypes. The variety of platforms used by the developers of these tools makes it necessary to define an overall technical architecture for establishing interoperability of the various software components.

The notion of interoperability that has been pursued in IMPACT is defined by ISO/IEC 2382-01, Information Technology Vocabulary, Fundamental Terms, as: "The capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units." Accordingly, the concept of interoperability in IMPACT is aimed towards the implementation of a highly flexible and easy to use technical framework and the integration of the software components within that framework. Interoperability of all tools as a main principle is herein achieved through two abstraction layers: individual software tools will be wrapped as web services, which are then again wrapped in a so called "basic" workflow module.

Web services are provided by describing the publicly available interfaces of the service using the Web Service Description Language (WSDL), and by using Simple Object Access Protocol (SOAP) for data exchange. Basic workflows are provided by wrapping the web services and their documented in- and outputs in a dataflow description used by the Taverna workflow management system, thereby exposing relevant features of the software tools in the form of ready-to-use components. The main advantages of this approach are the guaranteed compatibility of the tools offered as web services – avoiding incompatibility issues between different web service frameworks – and the reusability of a web service implementation.

This report is basically divided into two major sections: the first one will detail the current status of work on the Interoperability Framework architecture, the software components it has been built from and the functionality it provides, whereas the second section will be describing how each of the tools and applications developed by IMPACT partners are envisaged to be integrated with the Interoperability Framework, thereby enabling interaction and data exchange between them.

2. Evaluation of IMPACT Interoperability Framework

2.1 Components

The Interoperability Framework constitutes a service-oriented architecture (SOA) based on mature open source software components supported by an active user community. The Apache Software Foundation is a highly active and reliable open source software development community. IMPACT has therefore adopted the strategy to choose Apache Software Foundation projects in the first line, and evaluate alternatives only if the default Apache Software solutions appear to be inappropriate or too complex for the IMPACT project purposes. By choosing open source Apache projects as the base layer for the IMPACT technical framework, it is envisaged that this will facilitate takeup in a digital library environment as well as reduce overall adoption costs and support post project sustainability of technical outcomes.

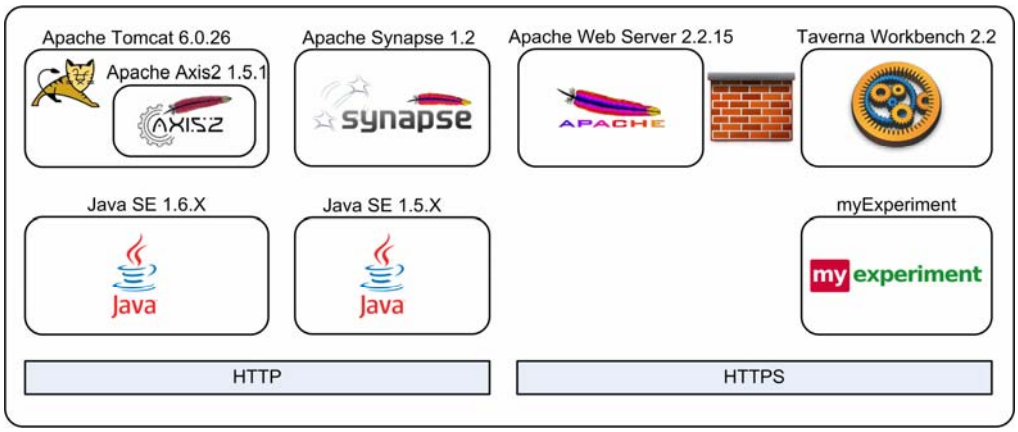

Figure 1: Interoperability Framework core software components

2.1.1 Software

The IMPACT Interoperability layer is implemented on the basis of the following technical components:

- Java SE Development Kit (JDK), JDK 6 Update 20 or higher; Java technology provides the technical ground for platform independence and is the main abstraction layer in the web service wrapper and framework.
- Java SE Development Kit (JDK), JDK 5 Update 22;
  Java 5 is additionally required to Java 6 for encryption since Apache Synapse (see below) uses a version of the Bouncy Castle Cryptography library which depends on this particular Java version. This will be replaced as soon as the legacy dependency is fixed.
- Apache Tomcat server Version 6.0.22 or higher;
  Apache Tomcat has been chosen as the servlet container for IMPACT services. It is open source, very widely used and provides a familiar environment to run java code.
- Apache Axis2 Java Version 1.5.1 or higher;
  Apache Axis2 Java Version was chosen as the web service framework over Apache CXF due to the fact that is very widely used, especially in connection with other Apache projects.
- Apache Ant Version 1.8 or higher;
  Apache Ant is used as the build tool for java projects in the project.

- Apache Synapse Version 1.2 or higher;
  Apache Synapse was chosen as it is lean and supports features of high importance to the IMPACT vision, such as load balancing and service failover (skipping services which for some reason are unavailable) for distributed web service endpoints.
- Apache Web Server Version 2.2.15 or higher (optional);
  Apache Web Server can be optionally used in the deployment of the Interoperability Framework as a proxy server to redirect to the application server Tomcat.
- Taverna Workbench Version 2.2 or higher;
  After initial evaluation of several candidates, two workflow systems were considered more thoroughly for workflow definition and execution: Taverna and Apache ODE. The main argument for Apache ODE was that as a software component from the Apache Software Foundation it is fully compliant with the other components in the Interoperability Framework. The drawback however is the complexity of BPEL used in Apache ODE. The main advantage of the Taverna system is that it provides an out-of-the-box solution for workflow design and execution. Also, it comprises an easy-to-use workbench, supplemented by a command line execution tool and a remote execution server that allows Taverna workflows to be run on other machines, web pages or as a service. Taverna uses an open standard XML format for workflow descriptions (T2FLOW/SCUFL2) that can be ported to BPEL, if need be. Therefore Taverna was chosen as the most suitable software for designing and executing IMPACT workflows.
- myExperiment virtual research environment
  The myExperiment environment, which is also part of the myGrid consortium and integrated with Taverna, has been selected for the workflow registry. The workflow registry constitutes the main platform for connecting the resources, such as tools and workflows, with the users in cultural heritage institutions throughout Europe. By means of this Web 2.0 platform users can discover and retrieve IMPACT workflows, share, rate and tag them and exchange their experiences in applying the tools in their specific context and with their material.

2.1.2 Benefits

From the IMPACT point of view, the Interoperability Framework architecture derived from the above components has beneficial characteristics in various aspects:

Modularity: Individual modules can be combined in a vast number of combinations, thereby enabling users to identify the most suitable processing chain for a particular kind of material being processed and guaranteeing the reusability of the components. In this respect, the service-oriented-architecture is the guiding architectural design principle; more specifically the principle of loose coupling of reusable processing units, minimising interdependencies between them. This not only allows providing IMPACT tools by means of different combinations, each specifically tuned to achieve optimal results on a particular type of input material, but in addition gives users complete freedom to rearrange certain steps or exchange individual modules with other ones that are available to them.

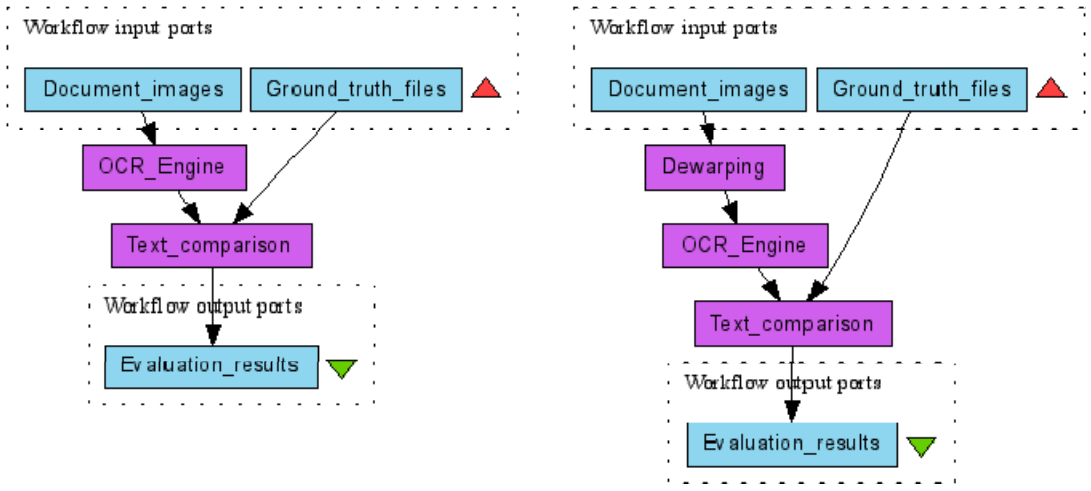

Figure 2: Workflows for assessing the effects of pre-OCR image enhancement (dewarping) on OCR results

Transparency: Each individual application/processing step can be tested and evaluated separately, so that it is obvious to the user whether a functional unit produces expected results and contributes to the overall quality of the workflow and what the cost is in terms of processing time. IMPACT tools can be run against their state-of-the-art counterparts or in a combination of different tools/services, thereby greatly supporting evaluation. For example if one module does not perform well on a particular type of material, it can be removed from the workflow or exchanged with another method simply by a few mouse clicks.

Flexibility: Due to the Interoperability Framework being platform-independent and capable of integrating many different types of software tools, automated workflows can be composed and rearranged whereas new workflows can be created and adapted from existing ones and the performance of these can be compared easily. Also, many components of the Interoperability Framework such as the Enterprise Service Bus or the web service framework can be easily replaced without requiring major changes to the overall architecture.

Extensibility: Third party components can be installed with very little extra effort. Therefore experimenting with and evaluating workflows is not restricted only to software tools developed during the IMPACT project. To demonstrate this, a number of free and open-source OCR and imaging tools have been integrated already and made available to project partners in dedicated workflows.

Open standards based: The Interoperability Framework is using only well-documented and actively supported open source software, mostly projects of the Apache Software Foundation. Thus, the Interoperability Framework can be reproduced and developed further by everyone interested, requiring no additional costs. All of the components used are freely available for Linux and Windows operating systems and also as open source code - the Apache License applies for all of the above Apache projects, and the LGPL for the Taverna/myExperiment parts (please refer to the according project website for additional information). Data exchange between software components is based on

widely used XML standards such as METS/ALTO for encoding of structural information and of the OCR recognised text, and SOAP as the message exchange protocol.

Accessibility: The Interoperability Framework can be accessed via different types of interfaces. There is a user-friendly, graphical workflow design and execution interface, a web client generator, which allows seamless integration into web sites, and a Java API.

Collaboration: The community-wide applicability and optimisation of workflows strongly depends on users actively working with them. IMPACT aims to make workflows accessible by various channels (including Web 2.0 features) to the stakeholders. In order to encourage people to actively collaborate, it is necessary to create easy-to-use workflows which are comprehensively described and documented. By utilising the myExperiment platform as the environment for community-based workflow design, IMPACT also contributes to the building of capacity in the wider digitisation community.

Scalability: The web service components, as the basic layer of the Interoperability Framework, are deployed in the IT infrastructure of different partner organisations in Europe. This creates a distributed network with cloned services available in a redundant way, which allows distributing the workload, avoiding single-point-of-failure (SPOF) and adding additional computing capacity whenever required.

2.2 Interfaces

On top of the Interoperability Framework architecture outlined above, various interfaces have been established for the demonstration work of IMPACT tools. These are described in more detail in deliverable *D-OC5.5 Demonstrator Platform with deployed tools and applications*.

Basically, there are two main user interfaces and a machine interface (API). The two user interfaces consist of
a) a locally deployable platform (Local Demonstrator Platform)
b) a web based platform (Central Demonstrator Platform)
Both of these interfaces use the various layers in the architecture of the Interoperability Framework, thereby accessing IMPACT tool representations through web services and workflows. For each of the IMPACT tools, at least a single web service and basic workflow fragment are created, comprising the web service with it's specific in- and outputs to be used with the Taverna workflow engine, plus some documentation. Based upon these workflow fragments, demonstrators can by dragging and dropping easily design comprehensive workflows for testing certain combinations of tools on various types of material. Working with the workflow fragments also relieves the demonstrators from dealing with technical issues relating to implementation of individual tools or the task of establishing data exchange between web services. It provides an additional abstraction layer that makes it much simpler to understand and apply combinations of tools and applications. The workflow and Taverna aspects of the Interoperability Framework also provide evidence to support the positive effect of cross-domain collaboration and effective technologies and approaches for distributed but domain dependent teams.

2.2.1 Local Demonstrator Platform

When it comes to direct user interaction with IMPACT tools, it is that the Taverna Workbench is utilised as a powerful means to compose and execute a chain of processes – a scientific workflow – in the IMPACT domain. Through the graphical user interface of Taverna users can quickly create and execute workflows thereby establishing a practical way to perform experimental workflow development. A number of screen casts have been produced which explain in detail the main functionalities of Taverna and its intended use in IMPACT: http://fue.onb.ac.at/impact/tg/vid/index.html.

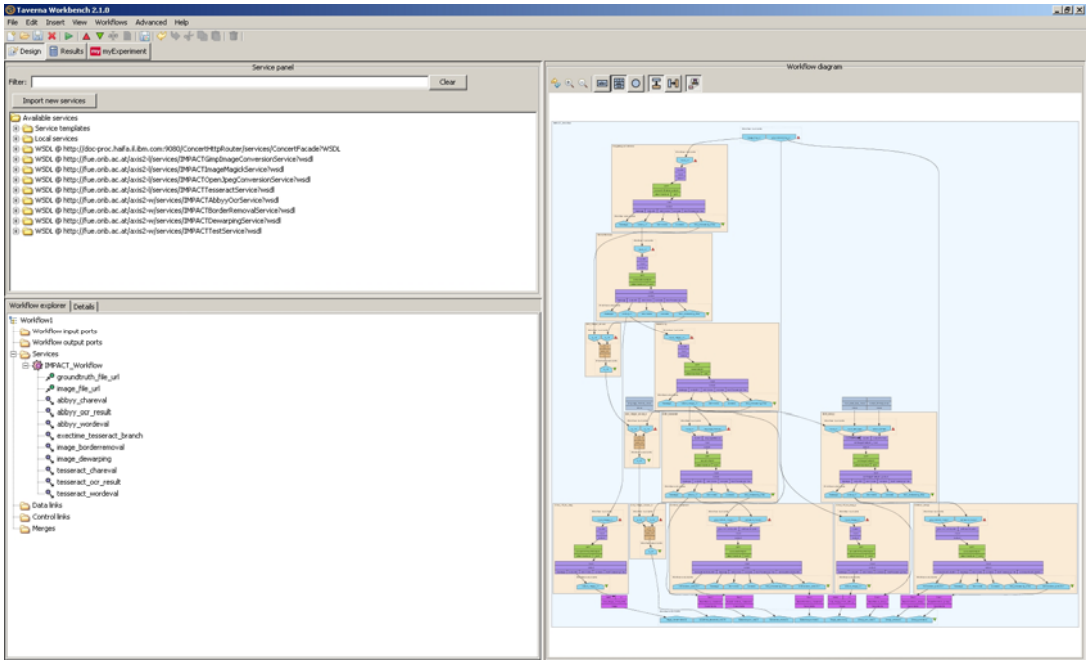

Figure 3: Design perspective for manipulating IMPACT workflows using Taverna Workbench

The Workbench interface consists of three windows (service panel, workflow explorer, workflow diagram). The service panel, in the top left corner of the Workbench, contains the available services that can be used in workflows. The workflow explorer, in the bottom left corner of the Workbench, contains the hierarchical tree view of the current workflow. The workflow diagram to the right contains the workflow diagram image. It can be used to modify the workflow and exposes several options for configuring the layout and display of the diagram: top to bottom orientation, left to right orientation, zooming in and out, showing all service ports or service as a black box, etc.

The perspectives (Design, Results, myExperiment) can be switched by clicking on the corresponding buttons underneath the main menu. The main focus lies on the design perspective of the Taverna Workbench which provides users with an easy way to build comprehensive workflows from individual tools. The results perspective shows the progress of a workflow run and the results obtained. Individual steps in the process and intermediate results they produce can be analysed by clicking at each of the steps. The myExperiment perspective allows registered users to access and retrieve workflows which have been registered at myExperiment from within the user interface.

Figure 4: Results perspective of IMPACT workflow using Taverna Workbench

2.2.2 Central Demonstrator Platform

IMPACT web services and workflows can also be executed directly via the project website and without the need to prior install any software on a local computer. In order to provide such functionality, the Taverna Server version 0.2.1 has been deployed on an Apache Tomcat container, and a client application developed for interacting with the server. Thereby a central access point for experiments with IMPACT tools and services has been created which will be particularly useful in showcasing functionalities of IMPACT tools to interested parties external to the project. The tools and applications which have been integrated through the website are available through the Central Demonstrator Platform, accessible with a user account from http://www.impact-project.eu/taa/dp.

The Central Demonstrator Platform currently provides access to IMPACT tools and resources by a number of ways:

- A platform to retrieve and execute applicable digitisation workflows from myExperiment for demonstration purposes. This component allows for the execution of Taverna workflows through a browser by parsing the uploaded workflow description file (T2FLOW) and then generating a dynamic HTML form accordingly. The workflow can then be executed remotely and the results will be presented to the user via the website. http://www.impact-project.eu/taa/dp/workflowclient/
- A platform to execute IMPACT tools which are available as web services. This component provides a simple way of accessing the individual web services by parsing a web service description file (WSDL) provided by the service registry and then generating a dynamic HTML form with the necessary parameters for executing the web service accordingly. http://www.impact-project.eu/taa/dp/webserviceclient/
- The web client of the CONCERT (COoperative eNgine for Correction of ExtRacted Text) tool. http://www.impact-project.eu/taa/dp/concert/

- The web client of the Functional Extension Parser.
  http://dea-gulliver.uibk.ac.at/org.dea.impact.FEP_Prototype.FEP_Prototype/FEP_Prototype.html
- The Named Entities Repository.
  http://www.impact-project.eu/taa/ee/tools/nerep/
- The Metrics Toolkit, a web application for statistical evaluation of the outputs of different workflows based on OCR results and ground truth.
  http://www.impact-project.eu/taa/mt/

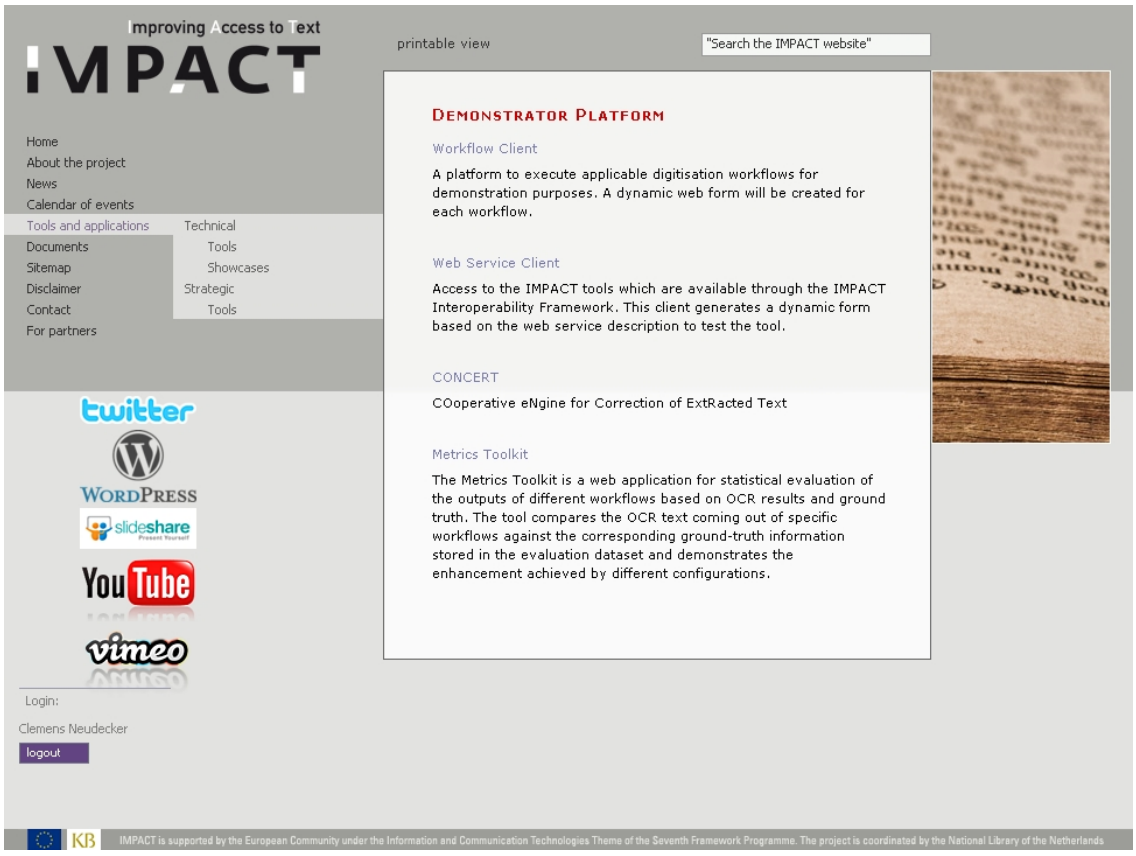

Figure 5: Demonstrator Platform integrated with the project website

2.2.3 Machine interface

The machine interface (API) is described by a WSDL document which defines the available operations for the service. The WSDL includes an XML schema document that strictly defines the data types that can appear in the SOAP requests and responses. Due to this, it is also possible to automatically access the functionality of the tools by means of any client utilising the SOAP protocol. As the SOAP requests and responses follow defined standards, almost any programming language with the appropriate library support can be used. Languages currently known to have this support include C++, C#, Java, Perl, Python and Ruby. Because the data is returned in a standardised machine-parseable format, it can be directly integrated into a third-party web site or application.

2.3 Repositories

A huge and varied amount of data is produced in the IMPACT project that needs to be stored, accessed and managed in a coherent and sustainable way. Several repositories have been set up to provide functionalities such as storage and retrieval of datasets (images plus ground truth files and metadata), source and object code as well as results of demonstrator activities.

2.3.1 Image and ground truth repository

An image and ground truth repository has been set up to manage the datasets collected from libraries. The datasets consist of high quality images selected by the content holding partners as being representative for their collections, a number of ground truth files for a subset of the images (which will be used in development and evaluation) and a set of metadata to describe and search for objects in the repository. It is used both by libraries to explore the images provided by other libraries and to contribute accordingly (ensuring that their contents are adequately represented) and by technical partners to identify material required as a baseline for development. The core software components of the image and ground truth repository are a high performance file server for the images, an SQL database for hosting the metadata and a web-based portal for accessing and managing the collection items. The image and ground truth repository is hosted by the University of Salford. A username and password are required for accessing the repository. It can be accessed from here: http://www.prima.cse.salford.ac.uk:8080/impact-dataset/

2.3.2 Software versioning repository

All software components and tools in IMPACT are made available through a central software versioning repository which is hosted by the University of Goettingen, thereby facilitating joint software development between technical partners. Apache Subversion is used as the version control system and a subversion client is required in order to download, modify, or upload software from the repository. A username and password are required for accessing the repository. It can be accessed from here: https://develop.sub.uni-goettingen.de/repos/impact/

2.3.3 Demonstrator results repository

Resulting data (images and text files) obtained through the execution of IMPACT web services and workflows by demonstrators are cumulatively stored on the servers running the Interoperability Framework. As demonstrators want to conveniently access this data for evaluation and reporting purposes, a dedicated demonstrator results repository was set up that exposes the required functionalities, such as querying, sorting or aggregating the data. The user requirements have been discussed jointly with the demonstrators and a simple web service implementation has been agreed. Accordingly, the results repository provides the means to store end results of demonstrator runs in a permanent and structured way, to query the processed material by metadata or workflow ID and to collect provenance information and the final outputs of a workflow run. The statistical outputs can be accessed as XML or XLS, thereby enabling demonstrators to aggregate and present results in the way required.

2.4 Guidelines

All technical partners have been encouraged to provide web services for their tools according to the guidelines that have been produced by OC5. Framework staff also supports technical partners in providing a generic web service wrapper solution based on the Interoperability Framework architecture components. The generic web service

wrapper which has been developed for that purpose is a Java-based program which uses the command line interface (CLI) of a native software component available for Windows or Linux. The wrapper is then provided as a web service with the interfaces required by the corresponding tool. The following diagram illustrates this concept:

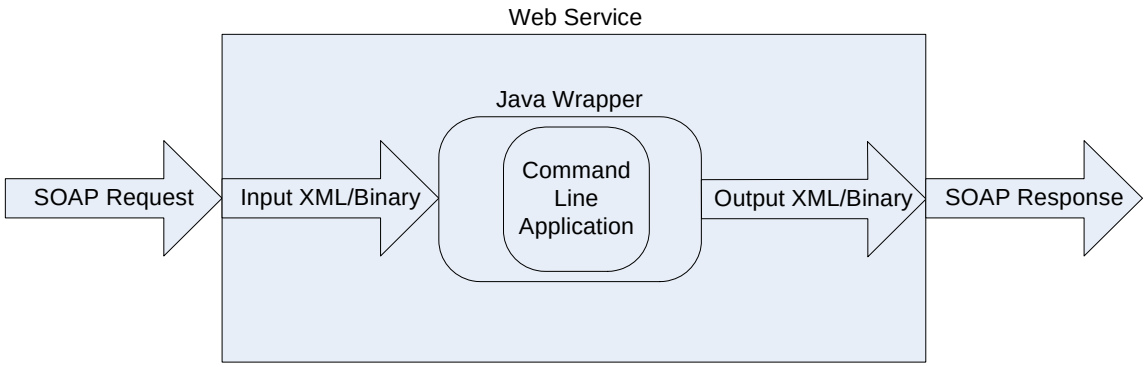

Figure 6: Generic web service wrapper

The main advantages of this approach are the guaranteed compatibility of the tools offered as web services – avoiding incompatibility issues between different web service frameworks – and the reusability of a generic web service implementation. As a consequence, as far as their contribution to the Interoperability Framework is concerned, technical partners can concentrate on additional development required for filling gaps which impede the intercommunication of the tools.

2.4.1 Guidelines for creating web services

Elaborate guidelines for software developers have been created, instructing them how to provide their tools as a web service using the Generic Web Service Wrapper. A simple web form for creating a skeleton project via the Generic Web Service Wrapper has been created and is available at http://fue.onb.ac.at/IMPACTServiceSkeletonGenerator. The Guidelines discuss in detail all aspects from the setting up of the environment required to build the web services to the usage of the Generic Web Service Wrapper as well as testing and customisation of the web service.

2.4.2 Deployment guidelines

In addition, user focussed deployment guidelines have been created, instructing demonstrators precisely how to deploy the Interoperability Framework and IMPACT tools and applications in a productive environment. The Guidelines describe step by step how to deploy the framework on a local server, how to install the tools and web services, how to do some initial testing and how to secure the web services by HTTP over SSL (HTTPS) and user authentication.

2.5 Formats

The Interoperability Framework supports two formats for representing textual content encoded in XML, ALTO and PAGE XML, both of which are explained in slightly more detail below. Other encoding formats such as ABBYY FineReader XML, Omnipage-XML, TEI, hOCR or plain text will be to some degree supported by integrating XSLT

style sheets for the necessary format conversion tasks within workflows. The target namespace for the IMPACT schema collection is accessible at: http://www.impact-project.eu/schemacollection.

2.4.1 ALTO

The ALTO (Analysed Layout and Text Object) standard is an XML schema that details technical metadata for describing the layout and content of physical text resources, such as pages of a book or a newspaper. It most commonly serves as an extension schema used within the Metadata Encoding and Transmission Schema (METS) administrative metadata section. Since August 2009 ALTO is officially maintained by the Library of Congress: http://www.loc.gov/standards/alto/alto-v2.0.xsd

ALTO was selected as the main output format for the results produced by the OCR tools in IMPACT. The main reason for this was that ALTO is very widely used among digital libraries and that IMPACT wants to meet the needs of this community. Nevertheless it turned out that ALTO in its current version does not meet all the requirements set out by the technical partners of the IMPACT project, i.e. some of the IMPACT tools provide more information than can currently be encoded within the ALTO format. Therefore a working group was formed in the project to provide suggestions on how the ALTO format could be extended and improved in the light of recent developments in OCR encoding. The working group has come up with 11 suggestions for changes and additions to the standard, most of them also explained with an XML code snippet. After some initial discussion with the ALTO editorial board, IMPACT has delivered a document detailing these suggestions to the ALTO technical subcommittee in May 2010 where they are currently being considered for a next format update.

2.4.2 PAGE

The PAGE (Page Analysis and Ground-truth Elements) standard is an XML schema image representation framework that records information on image characteristics (image borders, geometric distortions and corresponding corrections, binarisation etc.) in addition to layout structure and page content. The suitability of the framework to the evaluation of entire workflows as well as individual processing stages has been extensively validated by using it in high-profile applications such as in public contemporary and historical ground-truthed datasets and in the ICDAR Page Segmentation competition series. PAGE is developed and maintained by the Pattern Recognition and Image Analysis (PRIMA) Research Group at the University of Salford:
http://schema.primaresearch.org/PAGE/gts/pagecontent/2010-03-19/pagecontent.xsd

The PAGE format has been chosen as the most appropriate format for encoding of ground truth after a very careful evaluation of the available options. It became clear that no other encoding format for OCR output is rich enough to provide the necessary structure and granularity to record all the information that needs to be contained in the ground truth files and leveraged by the evaluation tools to cover the variety of challenges and issues the IMPACT tools will tackle. Due to the complexity of such ground truth, a highly efficient tool for creating large amounts of data in PAGE format was needed. In the Aletheia tool, this has been produced and will become available as an additional project outcome.

2.6 Testing

Various functionality tests are carried out by tool developers, with any bugs and errors spotted being recorded in a central issue tracker. Unit tests are performed for every individual tool using the command line test client before

registering the web service with the service registry. Once basic functionality has been verified, integration testing is done to determine whether there any issues with regard to interoperability, security or scalability of the module. Additional input/output tests are executed by creating sample client requests and analyzing the results. Binary data transfer is tested for images up to 350 MB file size. Finally, the tool/web service is executed against a defined test bed and usability is assessed by the demonstrators in the libraries (user testing) before final sign off.

For the software components powering the Interoperability Framework, functional testing and monitoring of the web services is done using the SoapUI tool. SoapUI provides a range of features for service simulation, functional testing and load testing. For example, detailed tests cases can be easily produced within the application and saved as an XML document. This XML document can then simply be uploaded to the monitoring service and will become available as a test scenario that can be immediately executed or integrated in more complex test scenarios. This greatly supports flexibility in testing and allows for the continuous tracking of performance of individual services or more comprehensive test cases. A number of such test cases have been produced and are executed on a regular basis by a job scheduler. The results obtained by these runs and the according information about the availability and performance of individual web service endpoints are recorded in an SQL database. An automated comparison of results from the periodic runs also provides a way for the validation of the platform as a whole when changing or updating individual components.

SoapUI load tests

Add a SoapUI load test project

Name	First endpoint	Created at	Monitoring		
My Load Tests	https://fue.onb.ac.at/axis2-1/services/IMPACTGimpImageConversionService	2010-08-04	on	test results	delete
bogus test	https://foo.bar/axis2-w/services/IMPACTBorderRemovalService	2010-08-03	on	test results	delete
test	https://fue.onb.ac.at/axis2-1/services/IMPACTGimpImageConversionService	2010-08-03	on	test results	delete

Recent errors

load test	test_step	miliseconds_min	miliseconds_max	miliseconds_avg	miliseconds_last	bytes	bps	tps	runs	errors_reported	ratio	date
bogus test	removeBorderByUrl - Request 1	3	1401	117.27	5	0	0	5.7	58	62	106	2010-08-03 18:00:31 UTC
bogus test	removeBorderByUrl - Request 1	2	1515	122.34	5	0	0	5.39	55	59	107	2010-08-03 15:20:32 UTC
bogus test	removeBorderByUrl - Request 1	2	1436	123.01	26	0	0	5.52	56	60	107	2010-08-03 15:01:32 UTC

Figure 7: Web service monitor

2.7 Challenges

While the concept of a service-oriented architecture yields several advantages for establishing interoperability between different software modules, it does also give rise to a number of challenges. The architecture contains several abstraction levels such as the wrapping of command-line tools into web services, the wrapping of web services into basic workflows etc. which introduces complexity in managing dependencies between those layers.

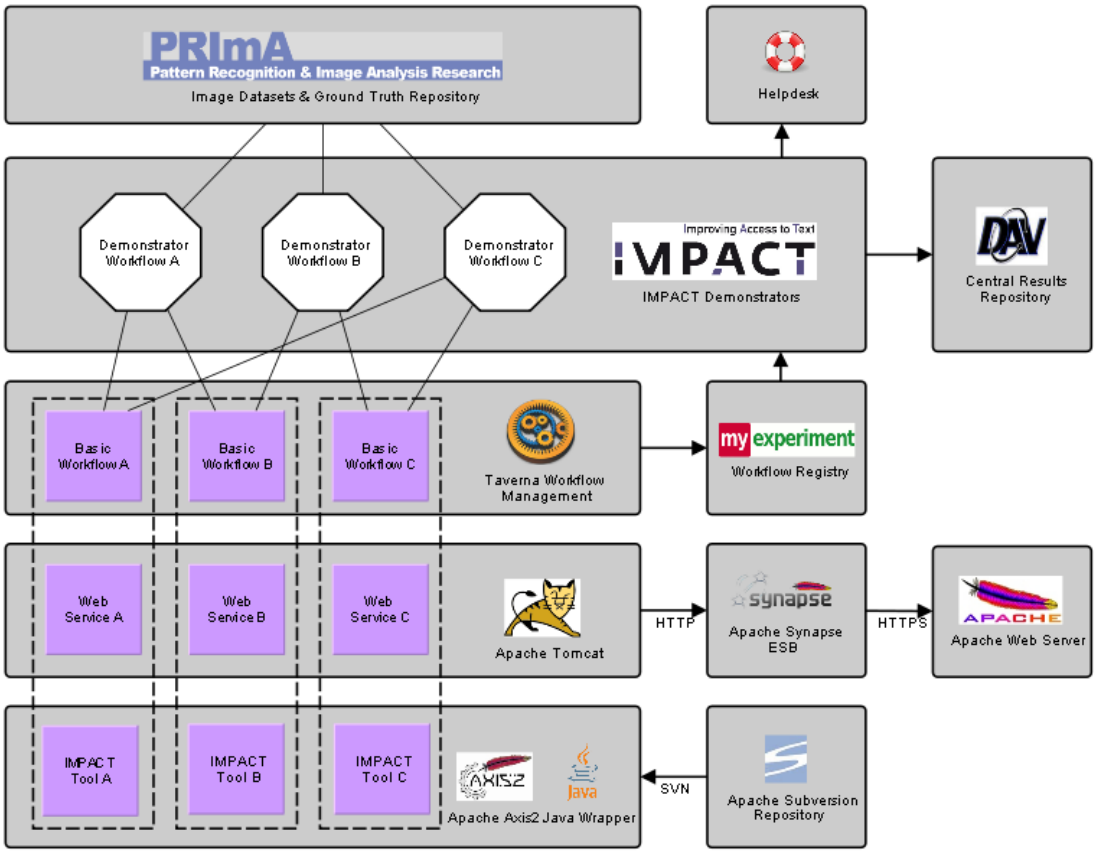

Figure 8: Complete overview of IMPACT technical framework interaction

Changes in the lower abstraction levels can have a high impact on the overall availability of the architecture. If not managed carefully, these changes can cause a partial or a whole malfunction. For example a new version of a tool might use different names for the input/output ports than the earlier version which would cause the related basic workflow fail to function. Or a service endpoint might move to another server which would have the same effect on the workflows that are built upon this service. The usage of basic workflows also limits the impact of changes to the service interfaces. However at some point there is a need to change the interface of the basic workflows and manage versioning without effecting service consumers. IMPACT has therefore decided to implement new releases of tools also as new web services and basic workflows. This ensures that
a) a version of the web service and workflow is always kept available that is known to be working and
b) it is relatively straightforward to create fresh mashups from new workflows, whereas it is a complex exercise to adapt all references to a certain tool or nested workflow within already existing mashups.

Another challenge for the architecture lies in its aptitude for large scale document processing. Usually masses of images are used in digitisation workflows (mass digitisation) which are rather big in size (up to hundreds of megabytes for a high quality uncompressed image file). Running workflows with thousands of images might cause an overhead in time, memory and/or bandwidth, which all affect the overall performance. As all of the IMPACT components operate somehow on data streams by either modifying them or extracting information, the web service

must therefore support binary data exchange between web services which is supported by the generic web service wrapper in two ways: Data can be passed "by reference" where the SOAP message contains a URL reference to a file, or it can be passed "by value" where the binary data is attached to the SOAP message using the MTOM standard for binary data transfer, the first being the default method used in IMPACT. In order to keep the data transmission time as short as possible, the architecture encourages the usage of URL references instead of binary attachments in SOAP messages.

Additional efforts have been taken to scale the services to the large amounts of data that can be expected in a mass digitisation environment. An Enterprise Service Bus is used to provide load balancing by distributing the workload across various proxies of the same services, which are deployed on different locations. Failover functionality is provided by monitoring the web services and skipping the endpoints that are not available for some reason. Whereas these features are highly desirable and significantly increase throughput performance by the Interoperability Framework, this also requires that the distributed web service endpoints are managed in way that allows it to pin down technical problems to a specific processing unit at any of the remote locations.

Also, a sufficient level of security had to be provided in the framework. For this reason, web service messages are transferred via HTTP over SSL (HTTPS) and basic authentication was introduced for all web services, which is the advocated WS-I method to provide web service security as stated in the WS-I Basic Profile 1.1. However, this also gave rise to a number of issues, such as Taverna not being able to access the load-balanced endpoints once they were behind encrypted proxy services. This was resolved by issuing out SSL certificates for each server exposing web service endpoints to the framework and then making these certificates available to Taverna.

3. Evaluation of integration status of tools and applications

Most IMPACT tools and applications will run on their own but also be able to be integrated into a larger system. Apart from a few stand-alone desktop applications (e.g. the OCR post-correction tools), all tools will support batch-processing of documents without requiring human interaction, thereby making them suitable for batch processing of large quantities of data through automated workflows and enabling interaction with other modules exposed by the Interoperability Framework. Generally, there is a one-to-one mapping from tools to web services, which means that one tool will also be described as a service in one WSDL file. Different types of functionality will be offered as different web service operations, and additional parameters are offered as parameters of the operation if required for higher level workflow creation.

In some cases, a more indirect approach to integration was necessary. This is constituted by making either the entire tool or some of its functionalities available to an OCR engine (in this case either ABBYY FineReader Engine or the Adaptive OCR Engine). This applies particularly for tools that deliver best results when being an integrated component of an OCR engine, e.g. the segmentation tools or the Error Profiler. Thanks to the integration of ABBYY FineReader Engine and the Adaptive OCR Engine with the Interoperability Framework, these tools will in turn also be available in the overall architecture.

The following section will explain in greater detail the status of web service and workflow integration of all technical deliverables in the project.

3.1 TR1 – Image Enhancement

The goal of this work package is to develop a set of software tools to enhance the quality of the scanned document (input image) in order to maximise the results of the subsequent steps of segmentation and OCR.

This work package produces the following technical deliverables:

D-TR1.1(i): Initial binarisation/colour reduction toolkits (M22) (20 pm)

D-TR1.2(i): Initial noise and artefacts removal toolkits (M22) (20 pm)

D-TR1.3(i): Initial geometrical defect correction toolkits (M22) (20 pm)

D-TR1: Image enhancement toolkit (M38) (34 pm)

3.1.1 Initial binarisation/colour reduction toolkits

The initial binarisation/colour reduction toolkit has been delivered as an integrated component of ABBYY FineReader 10 Engine beta (source code) in January 2010. The beta has been integrated with the IMPACT Enhancement and Segmentation Platform for evaluation purposes. Due to a number of constraints with the beta, it has not been integrated with the Interoperability Framework yet. This is expected to happen in September 2010 when a next, more mature beta version will become available.

ABBYY FineReader Engine has an extensive API which generally allows easy integration with other tools. This means that by calling the binarisation method from the SDK, the derived bitonal image can be made available also to other OCR engines or tools available in the Interoperability Framework for further processing or evaluation of binarisation methods. Planned additional improvements of the binarisation modules in the final release of ABBYY FineReader Engine 10 will not affect the external API so that they can be easily integrated into the already existing tool chain.

3.1.2 Initial noise and artefacts removal toolkits

The aim of this toolkit is to detect page borders in order to be cropped. These unnecessary image regions may be parts of the sides of the book, parts of an adjacent page, additional noisy scanning area etc. surrounding the regular page area in a document image. A first initial version of the border removal toolkit was delivered in June 2009 in the form of a compiled executable. This version was integrated as a web service and basic workflow. The toolkit has also been integrated in the IMPACT Enhancement and Segmentation Platform for evaluation purposes. Since the release of the initial version of the border removal toolkit, three updates have been delivered:

(a) an improved algorithm released in January 2010

(b) a version in February 2010 which included several bug fixes and

(c) a greatly improved version in June 2010 that allows not only b/w but also greyscale images as input and delivers a greyscale image as output with noise from borders removed. Furthermore, the latest version of the border removal toolkit now also permits page cropping of two-page document images as a separate feature.

In order to allow an easy integration with the Interoperability Framework, during the development of the border removal toolkit the following specifications have been taken into account:

(a) the user does not have to set any parameters (a parameter-free methodology was involved),

(b) a great variation of document image layouts and scanning settings should be supported,

(c) images of any type (b/w, greyscale, colour) and format (TIFF, JPEG etc.) should be supported.

3.1.3 Initial geometrical defect correction toolkits

A primary goal of this task is to provide geometrical correction for page curl defects typically occurring close to the spine of a book. After having collected a first initial dataset, it became obvious that apart from typical page curl defects, another geometrical defect could be discovered frequently in the images, namely arbitrary warping. Other than page curl, arbitrary warping leads to curved lines not only in the spine but across the whole page. In order to tackle this eminent problem, work on geometric defect correction tools was divided into two separate tools, one specifically for page curl correction and one additional tool tackling specifically the issue of arbitrary warping.

An initial version of the correction tool for page curl was delivered in January 2010 in the form of a compiled executable. This version was integrated as a web service and basic workflow. The toolkit has also been integrated in the IMPACT Enhancement and Segmentation Platform for evaluation purposes.

Since the release of the initial version of the page curl correction toolkit, two additional updates have been delivered:

(a) a version in February 2010 which included several bug fixes and

(b) a major update in June 2010 that allows not only b/w but also greyscale images as input and delivers a greyscale image as output.

Furthermore, the latest version of the page curl correction toolkit permits the possibility to choose between two options: i) using two consecutive dewarping stages (coarse and fine), ii) using only a coarse dewarping stage in the case of documents with dense layout. In order to allow an easy integration with the Interoperability Framework, during the development of the page curl correction toolkit, the following specifications have been taken into account:

(a) the user does not have to set any parameters (a parameter-free methodology was involved),

(b) images of any type (b/w, greyscale, colour) and format (TIFF, JPEG etc.) should be supported.

3.2 TR2 – Segmentation

The aim of this work package is to go beyond state-of-the-art techniques in document image segmentation in order to efficiently detect and classify homogeneous regions in historical machine-printed documents and further process text blocks for detecting text-lines, words and, finally, characters that will feed an OCR classifier.
This work package produces the following technical deliverables:
D-TR2.1(i): Initial block segmentation and classification toolkit (M22) (19pm)
D-TR2.2(i): Initial text-line and word segmentation toolkit (M22) (19 pm)
D-TR2.3(i): Initial character segmentation toolkit (M22) (20pm)
D-TR2: Segmentation and classification toolkit (M38) (58 pm)

3.2.1 Initial block segmentation and classification toolkit

The initial block segmentation toolkit has been delivered as an integrated component of ABBYY FineReader Engine version 10 beta (source code) in January 2010. The beta has been integrated with the IMPACT Enhancement and Segmentation Platform for evaluation purposes. Due to a number of constraints with the beta, it has not been integrated with the Interoperability Framework yet. This is expected to happen in September 2010 when a next, more mature beta version will become available.
ABBYY FineReader Engine has an extensive API which generally allows easy integration with other tools. This means that by calling the block segmentation method only from the SDK, the results can be made available also to consecutive segmentation processes (text-and-line segmentation) or OCR engines such as the Adaptive OCR Engine. To further on achieve loose integration with other tools, a converter has been produced which takes recognition results of ABBYY FineReader Engine and exports them as a PAGE XML file. The PAGE XML format used as an interface between different components of the Interoperability Framework was not discussed in detail before adoption, so there were some issues with different representation of the same data which have been resolved by June 2010. Planned additional improvements of the segmentation modules in the final release of ABBYY FineReader Engine 10 will not affect the external API of ABBYY FineReader Engine so that they can be easily integrated into the already existing tool chain.

3.2.2 Initial text-line and word segmentation toolkit

Text line segmentation and the following word segmentation are the subsequent steps and therefore depend on the results from page (block) segmentation. Accordingly, information about found regions had to be made available in a common format. This problem has been solved by the PAGE XML exporter provided by ABBYY. There still are some minor open issues such as the accuracy of region outlines (should be polygons instead of boxes) and overlapping regions which violate the requirements on a valid segmentation result.
A compiled executable for Windows operating systems (including required third party DLLs) and a batch file to run the tool in all modes (text line segmentation, word segmentation, component based method, profile based method) was delivered in January 2010. The command-line tool can be controlled by a set of parameters and therefore easily integrated using the generic web service wrapper. However, as this requires the results from the block segmentation to be available, integration has been postponed until this is the case. The toolkit has been integrated in the IMPACT Enhancement and Segmentation Platform for evaluation purposes.

3.2.3 Initial character segmentation toolkit

In this part of work efficient techniques are developed for word segmentation into characters in order to feed them to an OCR classifier. A first initial version of the character segmentation toolkit was delivered in June 2009 in the form of a compiled executable. Supportive material of this version consists of:

(a) a converter in order to accept PAGE XML files as input and

(b) a viewer for character segmentation results.

The first update (second version) of the character segmentation toolkit has been delivered in June 2010, incorporating the changes necessary to make it fit for integration with the Adaptive OCR Engine. The character segmentation toolkit uses as input the word segmentation coordinates that are produced by the ABBYY FineReader Engine corresponding module and produces several character segmentation variations, each one provided with a confidence score. These segmentation variations together with the segmentation variations produced by the Adaptive OCR character segmentation toolkit will be used as input to the Adaptive OCR Engine. In order to allow an easy integration with the Interoperability Framework, the architecture specifications of the Adaptive OCR Engine have been followed and the input/output of the character segmentation module adapted accordingly. In order to integrate with the Adaptive OCR Engine, several character segmentation variations had to be produced as well as the suitable XML encoding of the result. The algorithm was modified in order to produce several character segmentation variations (each one having a confidence). Currently, the testing of the integration of the character segmentation toolkit with the Adaptive OCR Engine is being performed. The character segmentation toolkit has been integrated in the IMPACT Enhancement and Segmentation Platform for evaluation purposes. As there is no subsequent step in the segmentation process, it is sufficient to integrate the character segmentation with OCR engines only. A web service and basic workflow are therefore not expected for this module.

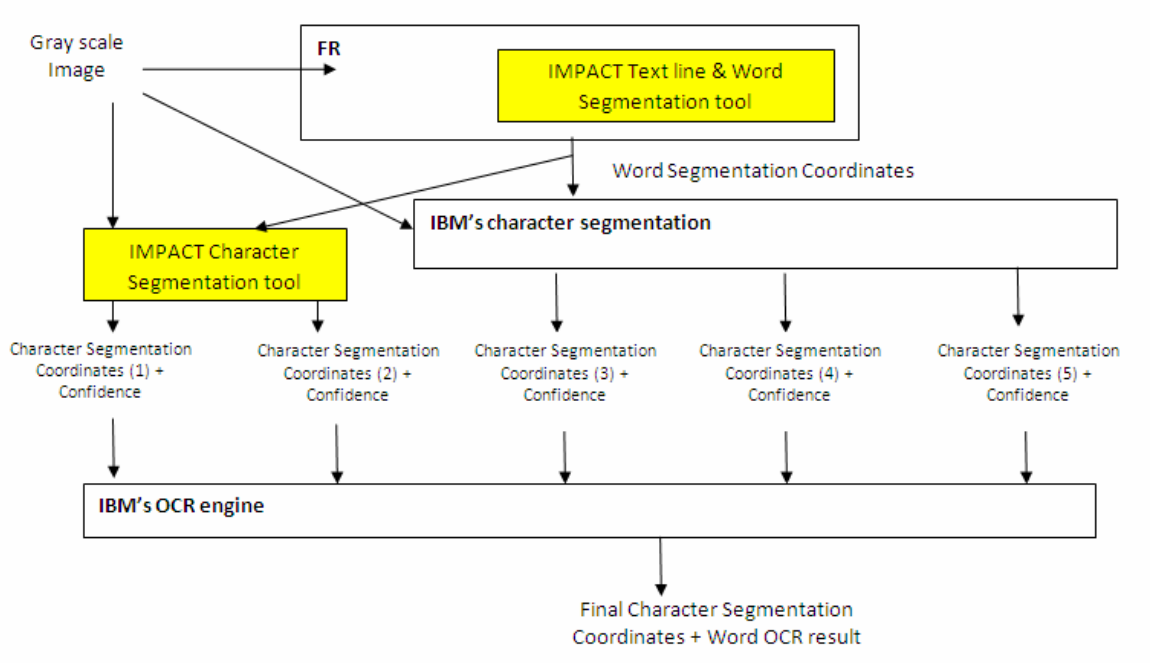

Figure 9: How to integrate TR2 segmentation toolkits to TR3 Adaptive OCR

3.3 TR3 – Adaptive OCR

The aim of this work package is to develop an automated approach for character training, which is referred to as Adaptive OCR. This system would learn from its own errors and automatically tune itself to the specific font character.

This work package produces the following technical deliverables:

D-TR3.2 Stand alone adaptive OCR module with partial functionality (M-12) (37pm)

D-TR3.3 Enhanced adaptive OCR module with full functionality. (M-24) (37pm)

D-TR3:4 Adaptive OCR system (M42) (49 pm)

3.3.1 Stand alone adaptive OCR module with partial functionality

This work package calls for the creation of a stand alone Adaptive OCR system. At an early stage of work it became clear that superior results can be achieved by adopting word level adaptive recognition (as opposed to conventional character level approach). This approach has been presented to the technical team and approved. The initial system has been tested on a defined benchmark where it achieved 23% improvement in the Figure of Merit (equivalent to 23% reduction in the manual correction load) as compared to the state of the art conventional OCR engine (ABBYY FineReader SDK Version 9) with optimal settings. By request of IBM, the API of ABBYY FineReader Engine has been extended to allow adaptive recognition cycles. The API now includes a possibility for training of the recognition engine on incorrectly recognised characters and the possibility to obtain correct shapes of recognised glyphs.

3.3.2 Enhanced adaptive OCR module with full functionality

From the initial prototype delivered in December 2008, the internal target was to have a version with full functionality at the end of 2009 and the web service version of the Adaptive OCR Engine and a basic monitoring implementation ready by July 2010. The last and stable version of the web service was delivered in June 2010.

A challenge was to create a more robust way to extract candidates from the ABBYY FineReader Engines results for font creation to be fed to the Adaptive OCR Engine. This is critical for the initiation of the Adaptive OCR mode. It was possible to improve significantly the candidate listing and ways will be explored in the remainder of the project to even do this automatically without any user intervention.

As envisaged in the Description of Work, the Adaptive OCR Engine integrates a number of software tools from other technical work packages. It implements the segmentation tools and utilises the historical dictionaries that are being created. In addition, the Adaptive OCR Engine itself has been integrated in the Cooperative Correction tool as an alternative OCR engine next to ABBYY FineReader Engine. Since being integrated also as a web service and workflow implementation, basically all other IMPACT modules available in the Interoperability Framework can interact with the Adaptive OCR Engine as web service.

3.4 TR4 – Experimental OCR Engines

The objective of this work package is to explore novel techniques and approaches to OCR processing which are highly promising from a research point of view.

This work package produces the following technical deliverables:

D-TR4.1 Inventory extraction prototype (M24) (17pm)

D-TR4.2 Typewritten OCR prototype (M24) (17pm)
D-TR4.3 Word spotting prototype (M24) (12 pm)

3.4.1 Inventory extraction prototype

This work package will introduce a novel technique that allows extracting a comprehensive inventory from a given document, i.e., a list of all characters used in the text. This technique is supposed to work without a-priori knowledge of the shape of the characters and of the language of the document, e.g. also in the case of non-Latin alphabets. Based on this inventory the allocation of characters to clustered patterns can be done easily. The result may also serve as input for the creation of new character templates.

Regarding the issue of integration, it is important to mention that the whole work package has been labelled as "experimental" from the beginning of the project. This is also reflected by the Description of Work, were it is stated explicitly, that the outcome will be a stand alone prototype implementing the experimental work. However, it needs to be emphasised that during the project the work plan was adapted so that the inventory extraction tool directly works together with the segmentation tools developed in the project.

In accordance with the recommendations from the second review report, it was decided to introduce several additional features and to transform this "experimental" approach into a full-featured OCR engine. As a first step, the tool will at the end of 2010 not only be able to cluster the glyphs of a document into classes of similar looking characters, but also add labelling functionality, i.e. the possibility to assign a Unicode symbol to each class by the user. To assist this process, several editing functions for the created clusters such as removing, merging or adding and removing instances to or from a cluster, are added in the graphical user interface of the tool. Thus, the work results in a tool that semi-automatically creates a flexible set of labelled characters for text documents. In a second step, the tool will be even further extended by a recognition functionality that is able to use the created labelled sets for the recognition of the given or other documents with similar font types. The functionality of the tool is therefore extended to a complete user-aided omni-OCR engine that is independent of any font type. The main application field will be digitisation environments, where documents with rare fonts occur as well as researchers who rapidly want to create a training set for certain documents. Additional optimisation and fine-tuning of the application and integration into Interoperability Framework is also scheduled while all of these activities that do extended the timeframe for the development are completely covered by the person-months originally allocated.

3.4.2 Typewritten OCR prototype

This work package will focus on developing a dedicated OCR engine for typewritten documents as they often occur in archival collections. Current document analysis approaches assume text of uniform intensity and quality and do not produce as high quality results as the new methodology proposes. In this approach, each character is isolated and enhanced individually. Broken characters are repaired by making a small number of hypotheses as to the location of the missing parts. A number of features are then extracted from each character image and, using a combination of classifiers, the character is recognised. The tool will be available as a compiled executable with the main functionality exposed via command line parameters. Input will be an image and a PAGE XML file; output will be a PAGE XML file enriched with text. Once segmentation results are available in form of PAGE XML files, the typewritten OCR prototype can be run in parallel or as an alternative to other OCR packages. This way, specific workflows can be set up tailored to the type of input material.

3.4.3 Word spotting prototype

The work package develops an alternative technique for historical document indexing based on spotting words directly on document images with the help of word matching while avoiding conventional OCR procedures. Word segmentation is first applied to all document pages and then spotting of the most interesting words (keywords) is applied directly on the document images.

A first version of the word spotting prototype was delivered in December 2009 in the form of a compiled executable. This version is a stand-alone Windows based software with a graphical user interface that requires human interaction. There have already been three releases of the word spotting prototype since. A significantly improved intermediate release was presented during the all staff meeting in Paris in April 2010 while the latest release was delivered in June 2010.

The word spotting prototype was not initially foreseen to be integrated with the Interoperability Framework but to operate as a stand-alone application being one of the outcomes in the experimental OCR engines work package. During the all staff meeting in Paris in April 2010 the possibility of providing a version of word spotting prototype without user interface was examined in order to investigate ways of integrating it also with the Interoperability Framework, thereby establishing interaction with other modules. A decision was taken to also provide a web service implementation of the tool with restricted functionality, therefore not requiring human interaction. In this way, there will be a direct integration with the overall IMPACT workflow which was not initially foreseen. For this additional effort only the resources that were already allocated to the work package will be used. This version of the word spotting prototype will be presented during the next all staff meeting in November 2010.

3.5 TR5 – Language Modelling and Dictionaries in OCR

This work package addresses the development of lexicon data and language models for text recognition in the context of historical documents.

This work package produces the following technical deliverables:

D-TR5.1 Text and error profiling module (M24) (20pm)

D-TR5.2 Post correction system (M36) (20 pm)

3.5.1 Text and error profiling module

The text and error profiling module is a software tool with documentation for profiling input documents recognised by OCR in terms of their language and in terms of the errors introduced by imperfect OCR. A first version of the text and error profiling module was delivered in December 2009 and integrated as a web service and basic workflow. In the course of 2010 and the development of the post-correction system, the text and error profiler will be integrated with this system as well. During the all staff meeting in Paris in April 2010, possibilities for integrating the text and error profiler with ABBYY FineReader Engine and the Adaptive OCR Engine have been debated. It has been agreed with partners IBM and ABYYY as well as with the sub project leader for TR that LMU will conduct an additional evaluation cycle on the basis of the suspicious character flag as provided by ABBYY FineReader Engine for characters that fall below a certain threshold of confidence. In addition to the output of the software as specified in the product descriptions and in the deliverable document, LMU will provide information on the suspicious characters in the OCR output. This information will be integrated into the XML output format of the tool. If the performance of the suspicious character flags can be improved in comparison with the ABBYY output, IBM and ABBYY agree to consider the

integration of the profiling tool into their OCR engines. The new functionality is foreseen to be implemented and evaluated until end of September 2010 and will be integrated into the web service in collaboration with the users of the service afterwards. The end vision is to use the profiling output exceeding mere OCR improvement as profiling of OCR outputs can also be used for subsequent applications as document presentation and information retrieval.

3.5.2 Post correction system

In this work package a system for post-correction of OCR results will be developed where dictionaries, text profiles, error characteristics and language models are fully interleaved to isolate misrecognised input tokens, and to find an optimal ranking order and confidence values for correction suggestions. Work on the post correction system started in 2010. The system will integrate the text and error profiling module as compiled code and is scheduled to be delivered in March 2011. The system will be a standalone desktop application which is intended to improve through human interaction OCR results as received from external service providers or contract partners respectively from internal digitisation centres.

3.5.3 External dictionary interface

ABBYY provides an interface for so-called “external” dictionaries which enables the integration of specific lexica developed by IMPACT language partners into the ABBYY FineReader Engine. A web service will be developed that is capable of running ABBYY FineReader Engine with an externally supplied lexicon. The choice of dictionaries will be limited to the set of IMPACT lexica present on the server. The service will be developed by wrapping a command line application compiled with ABBYY FineReader Engine.

3.6 EE1 – Collaborative Correction

The goal of this work package is to create a web-based platform that will facilitate users to collaboratively validate and correct OCR results.

This work package produces the following technical deliverables:

D-EE1.3 Full web-based system including input management. (M-24) (44 pm)

D-EE1.4 Enhanced system including user monitoring (M-36) (43pm)

3.6.1 Full web-based system including input management

The tool will minimise human intervention required for correction of OCR results and repurpose manual effort for additional tasks such as training the OCR engine and adding words to its vocabulary. The collaborative correction system has been delivered as a web client in November 2009 and as a web service suitable for integration with the Interoperability Framework in March 2010. Already several new versions have been released during the first half of 2010, adding features such as web services for loading books also into the Adaptive OCR Engine and to perform omni-OCR operations as well as a number of bug fixes and back-end process improvements.

A main challenge was to change the web service functions to be easy for external integration. Special care was given to use detailed structures for input to the system which reflects the functionality of the service. This way it is possible to avoid any internal knowledge about the input and output parameterisation. The tool will support at least three languages until the end of 2010 and at least additional three languages by the end of 2011.

3.7 EE2 and EE3 – Lexicon Structure, Tools and Content

These work packages provide general tools and guidelines for lexical data development from historical source material, the actual lexicon content and tools to deploy the lexicon in enrichment as well.
The work packages produce the following technical deliverables:
D-EE2.2 Named entities repository and collaborative environment (M12) (8pm)
D-EE2.3 Toolbox for named entities variant resolution, matching and classification (M24) (18 pm)
D-EE2.4 Practical guidelines and toolbox for building lexicon content (M24) (46 pm)
D-EE2.5 Toolbox for lexicon deployment in enrichment (M21; with decompounding: M36) (39 pm)
D-EE2.6 Final release of D-EE2.3 and D-EE2.5 (M48) (4pm)

3.7.1 Named entities repository and collaborative environment

The named entities repository is a central database in which project partners share named entity data for named entity lexicon building. The named entities repository was delivered and integrated with the project website already in 2009. Access to the named entities repository is possible with a website user account from: http://www.impact-project.eu/taa/ee/tools/nerep/ .

3.7.2 Toolbox for named entities variant resolution, matching and classification in historical documents

This toolbox supports extraction and classification (person, location, organisation) of named entities in historical documents and variant-independent retrieval (named entity matching). The first version of the named entities toolbox was delivered in January 2010. It features an adapted version of the Stanford named entities recognition tool. The named entity matching tools will be delivered before the end of 2010.
Named entity tagging of historical Dutch texts (1750 - 1940) will be made available as a web service by March 2011. The web service will allow named entities tagging with
1) Plain text input and output
2) XML input and output, provided that an element is indicated in the XML which contains the “plain text”. Content within this element will be tagged with named entity information.
Although the named entity tagging process is trainable and largely language-independent, it is not intended to deliver trainable named entity tagging as a web service. This step is part of a workflow to prepare documents for retrieval instead.

3.7.3 Practical guidelines and toolbox for building lexicon content

The toolbox for building lexicon content has been delivered in January 2010. It contains web-based user-interface tools for lexicon construction, but due to the nature of the tools there will be no web service implementation.

3.7.4 Toolbox for lexicon deployment in enrichment

A specific problem for historical material is the combined problem of variation and inflection. Accordingly, historical inflected word forms need to be reduced to their corresponding modern lemma form. Lemmatisation can be seen as the final step in the workflow for a document before it is indexed for retrieval. A typical workflow would then consist of OCR, named entity tagging and lemmatisation.
Lemmatisation for Dutch will be deployed as a web service. This web service will use, besides lexicon content,

lemmatisation components to reduce word forms to lemma form. The initial toolbox for lexicon deployment has been delivered in January 2010 and the web service will be delivered by March 2011.

3.7.5 Lexicon content

Lexicon content is used within the project:
1) to improve OCR by deployment within ABBYY FineReader Engine by means of the external dictionary interface and by deployment within the Adaptive OCR Engine
2) to support variant-independent retrieval on historical document collections
Deployment within ABBYY FineReader Engine is achieved by means of the external dictionary interface. Deployment for retrieval purposes is achieved by the named entity tagging and the lemmatization services. To ensure interoperability of the lexicon content on a broader scale, an appropriate LMF XML export has been developed.
A first version of the core lexicon of Dutch was delivered in February 2010, a first version of the core lexicon of German was delivered in January 2010.

3.8 EE4 – Functional Extension Parser

The main objective of this work package is the development of a Functional Extension Parser which will essentially increase the value of text recognition processes. Instead of getting "only" a good-quality text, libraries will get a functionally enriched digital object which can be used in a variety of ways and for a variety of different purposes. The proposed approach has the advantage that it directly utilises OCR output files so that it is independent from any segmentation, layout analysis software as well as from a specific OCR engine.
This work package produces the following technical deliverables:
D-EE4.2 Prototype: Functional Extension Parser (M24) (20 pm)
After having finished the original work plan in 2009, the work package was continued and extended. The final deliverable will be:
D-EE4.3 Functional Extension Parser – Productive Version (M48)

3.8.1 Prototype: Functional Extension Parser

An initial prototype of the web client for the Functional Extension Parser was presented during the all staff meeting in Venice in September 2009. In the original Description of Work, the work package was designed to provide a prototype only which was delivered after 24 months.
According to the extension, the main additional objectives for year 3 and 4 are the following:

1. To integrate the Functional Extension Parser module into the Interoperability Framework so that it is able to interact with other software modules developed in the project.
2. To enhance the rule set in order to also detect table of contents entries, headlines, running titles, footnotes, illustrations, captions, catch words and marginalia.
3. To transform the prototype into a productive version theoretically capable of processing millions of page images and to offer it as a licensed service to libraries and third parties after the end of the project.

The Functional Extension Parser will be made available after the end of the project as a commercial service provided by the Department of Digitisation and Digital Preservation of the University Innsbruck Library. The Department already provides commercial digitisation and OCR services to other cultural heritage institutions in the framework of

the European eBooks on Demand (EOD) Network. The Functional Extension Parser will also be hosted as a running service via the Interoperability Framework for at least 24 months after the end of the project without charging any costs to the IMPACT Centre of Competence.

In order to allow an easy integration of the Functional Extension Parser into the Interoperability Framework some consultations where made between the members of the technical teams regarding the interfaces required (e.g. considering the appropriate format for in- and output data, and the most suitable form of communication). It was decided to use a combination of METS/ALTO as common format for the in- and output of the system. The bundle of web services which will be provided by month 36 will consist of methods for the following use cases:

a) Authentication / login

b) Start the analysis

c) Get the current status of the analysis

d) Get the results of the analysis

Initially it was intended to use an asynchronous way of communication between the Functional Extension Parser web services and the Interoperability Framework. An asynchronous communication has the advantage that the client does not have to wait for the response of the client in order to proceed with the computations. This fact was important because the estimated time for the analysis may lead to timeouts on the client side. Tests with a dummy implementation have shown though that asynchronous web services can not properly interact with Taverna, the technology used by the Interoperability Framework. To overcome this problem it was decided to simulate an asynchronous communication by using a couple of synchronous web services.

3.9 OC3 – Evaluation Tools and Resources

The aim of this work package is the development of an application-oriented metrics toolkit and related resources for holistic system-level benchmarking, the creation of a representative test dataset with ground truth and the specification and development of a set of procedures and tools to support fully or partially automated performance measurement.

This work package produces the following technical deliverables:

D-OC3.3(i): Evaluation tools (M10) (25pm)

D-OC3.4: Metrics toolkit (test material and documentation on use and interpretation of results (M12) (10 pm)

3.9.1 Evaluation tools

The evaluation tools package comprises a number of individual tools tailored to specific tasks such as evaluating performance of binarisation, border detection and removal, segmentation and of course measuring the accuracy of OCR results. Most of these were initially foreseen to be used only by tool developers for performing necessary evaluation during tool development. However, the maturity of some of the tools and the high impact for automated evaluation of IMPACT tools by means of workflows led to the decision to also integrate several of the evaluation tools with the Interoperability Framework whenever technically feasible and of avail. A first version of the evaluation toolkit was delivered in December 2009 as a set of and has been integrated in the IMPACT Enhancement and Segmentation Platform for evaluation purposes.

As the evaluation tools suite encompasses a number of highly specific tools, possible effects of and efforts towards integration will need to be discussed individually.

a) Document image binarisation is a very important step in the document image analysis and recognition pipeline as the performance of a binarisation algorithm directly affects the recognition process. Therefore, it is imperative to have an evaluation toolkit which will account for the performance of the binarisation. The Binarisation Evaluation Toolkit that has been developed for this purpose offers many conveniences such as selecting areas of an image to construct the ground-truth and perform the evaluation, adjusting parameters to achieve estimated ground-truth of higher accuracy in difficult cases, double-clicking shortcuts, choosing the colour for the manual skeleton correction which is suitable for 24-bit colour images and more. As far as the evaluation stage is concerned, all the error measures that complement the recall and precision measures, such as broken and missing text, false alarms, component enlargement and merging, are shown at a different panel. Moreover, a report file is created in Microsoft Excel format that summarises the performance evaluation results along with recall and precision result images.
The toolkit was not selected for integration as the creation of binarisation ground truth is a very complicated task involving a lot of human labour so that only very small amounts of this type of ground truth can be produced during the project. Accordingly, the work that would be needed for integrating the tool can not be justified.
b) Border detection and removal is an important pre-processing step in order to obtain document images containing only the actual page without any surrounding areas introduced during scanning. The Border Detection Toolkit serves the purpose of evaluating the performance of border detection methods. It takes as input two document images, one being the original image and the other being the output of a border detection method, plus a ground-truth file, which represents the manually-input perfect layout of the document. The tool then compares the result against the original document image and outputs metrics representing the performance of the border detection method in question. As there will be only very little amounts of border removal ground truth produced and not many images in the datasets collected show this defect, there was no particular need for immediate integration. The tool may still be considered for integration it with the Interoperability Framework a later time though if there is a use case for it.
c) The Visual Comparison Tool is merely intended as a helper tool for human operators in judging visually about the quality of images and the effect of methods applied to them by means of a side-to-side comparison. As this will naturally always have to involve human interaction and is applicable only for a limited range of purposes, the tool was not considered for integration.
d) The Segmentation Evaluation Toolkit provides a unified framework for evaluating the performance of page segmentation at the region, text-line, word and glyph levels. It takes as input a segmentation file, which is the output of a segmentation method on a document image, and a ground-truth file, which represents the manually-input perfect layout of the document. The tool then compares the segmentation against the reference file and outputs statistics and metrics representing the performance of the segmentation method in question. The tool is provided in the form of a graphical programme for the Windows operating system, leaving to the user the ability to define scenarios containing specific penalties for segmentation faults. A first version was made available to developers already in October 2008. An updated version was released in May 2009. The update was mainly necessary to reflect extensions to the ground truth format used.
Due to the high importance of correct segmentation results in the results of an overall IMPACT workflow, it has been decided to also produce a command line application version of the tool which will take segmentation scenarios as an additional input and therefore will be suitable for integration in the Interoperability Framework. The command line version for integration via web service is scheduled for August 2010. This release will constitute a major step and will

allow full featured integration via a web service. To make this possible, the segmentation evaluation tool had to be modified from a stand-alone GUI application (to be used by specialists only) to a scriptable command line tool (to be integrated via the web service wrapper). A profile editor had to be implemented in order to create and save profile files which are used by the evaluation tool to set weights and penalties according to specific evaluation scenarios. A detailed XML Schema for evaluation results (raw data, statistics, and metrics) had to be developed together with an output function to store results in form of XML files compliant to this specification. As integration of evaluation tools was not foreseen in the original Description of Work, resources had to be allocated in order to achieve this new goal. However, the benefits of the additional integration efforts are considered very much worthwhile as starting from ground truth and segmentation files in PAGE XML it will be possible to obtain very detailed results regarding the performance of segmentation tools in specific scenarios. The tool is expected to be ready in August and integration will be tackled in September 2010.

e) The OCR Evaluation Toolkit provides a framework for evaluating the performance of a text recognition system. It takes as input two text files which comprise the correct transcription of the document image obtained by re-keying (ground truth) and the transcription produced from the OCR system. The tool then calculates various evaluation metrics on the character as well as on the word level. The tool is provided in the form of a graphical programme for the Microsoft Windows operating system. Due to the apparent importance of such a tool to libraries who want to evaluate the quality of their overall workflow or the IMPACT workflows, it became clear that integration of this tool as a web service would entail major benefits. A first version was released in October 2009 and immediately integrated with the Interoperability Framework. A renewed and improved version was released in June 2010, now introducing statistics output in XML format suitable for further processing and also supporting UTF-8 encoded text files.

3.9.2 Metrics toolkit

The Metrics Toolkit is a web application for statistical evaluation of the outputs of different workflows based on OCR results and ground truth. The tool compares the OCR text results coming out of specific workflows against the corresponding ground-truth information stored in the evaluation dataset and demonstrates the enhancement achieved by different configurations. The Metrics toolkit directly builds on the workflow layer of the Interoperability Framework, exposing its capability to compare different workflows with regard to their suitability for particular source material and presents the user with statistical information gathered from the available evaluation tools. It can therefore be considered an integral part of the Interoperability Framework architecture.

3.10 Other

For purposes of testing and evaluation against existing state-of-the-art methods, to prove the integration capabilities of the Interoperability Framework, and due to the simple necessity of e.g. format conversion tasks, a number of third party applications have also been integrated with the Interoperability Framework and made available as web services and basic workflows.

3.10.1 ABBYY FineReader Version 9

In order to have a baseline towards what can be considered the state-of-the-art in OCR technology without any improvements from IMPACT, ABBYY FineReader Engine Version 9.0 was kindly provided by ABBYY.

3.10.2 ABBYY to PAGE XML Exporter

As the segmentation tools as well as the segmentation evaluation tool require recognition results to be in PAGE XML format, an exporter was created to provide this output from ABBYY FineReader Engine.

3.10.3 Tesseract OCR

The Tesseract OCR engine was one of the top three engines in the 1995 UNLV accuracy test and is probably still one of the most accurate open source OCR engines available.

3.10.4 OCRopus

OCRopus is a state-of-the-art document analysis and OCR system, featuring pluggable layout analysis, pluggable character recognition, statistical natural language modelling, and multi-lingual capabilities.

3.10.5 OCR Transformation and Extraction

To enable evaluation of already existing OCR results from content holding partners, a service was needed to transform these into formats that are supported by the OCR Evaluation Tool, e.g. converting ALTO to TXT. A web service was built to comprise all the necessary format conversion tasks.

3.10.6 OpenJPEG Conversion

The OpenJPEG library is an open-source JPEG 2000 codec written in C language. It has been developed in order to promote the use of JPEG 2000, the new still-image compression standard from the Joint Photographic Experts Group (JPEG).

3.10.7 GIMP Image Conversion

GIMP is a versatile graphics manipulation package.

3.10.8 ImageMagick Image Conversion

ImageMagick is a software suite to create, edit, and compose bitmap images. It can read, convert and write images in a variety of formats.

4. Target integration milestones

A lot of emphasis has been put on the Interoperability Framework in year three of IMPACT. Accordingly, integration of already available initial toolkits and prototypes did proceed well and a number of important milestones have been achieved, such as the implementation of the Adaptive OCR Engine via a web service, the integration of the segmentation tools with the Adaptive OCR Engine or the newly identified options for including experimental OCR engines with the overall IMPACT workflow, to mention only a few.

In the next section, a number of additional targets are set out to regularly check progress against while working towards a more comprehensive integration of technical deliverables and a fully featured IMPACT framework. The near goal is to be able to demonstrate at the end of 2010 the early benefits that can be obtained by utilising an end-to-end workflow including all of the available technical modules from IMPACT.

June 2010 (M30)

- A web service and basic workflow are available for all initial image enhancement toolkits
- The Adaptive OCR Engine is integrated with the Collaborative Correction application web service, thereby establishing integration with the overall Interoperability Framework

September 2010 (M33)

- A web service and basic workflow are available for the Segmentation Evaluation tool
- An evaluation is done for deciding whether initial segmentation toolkits and the error profiler are suitable for integration with the Adaptive OCR Engine
- All experimental prototypes explore ways for integrating (parts of) their functionality with the Adaptive OCR Engine and the Interoperability Framework

December 2010 (M36)

- An overall integrated workflow for Taverna is available that includes the following processing steps:
  → retrieve images and ground truth files from the image repository
  → process the images with all/a combination of image enhancement tools
  → execute ABBYY FineReader Engine or Adaptive OCR with integrated segmentation toolkits
  → use the available lexical resources for improving recognition results
  → leverage the available tools for post-processing (such as the Functional Extension Parser)
  → evaluate final results against ground truth with regard to segmentation *and* OCR accuracy
  → store and make available final results, statistics and provenance information in the results repository

March 2011 (M39)

- All final image enhancement toolkits are integrated with the Interoperability Framework
- All final segmentation toolkits are integrated either with the Adaptive OCR Engine, ABBYY FineReader Engine or both
- All experimental OCR engines are integrated either with the Interoperability Framework, the Adaptive OCR Engine or both (only if integration efforts are justified by a significant improvement of functionality and/or results)

- All initial lexical resources can be used with ABBYY FineReader Engine and the Adaptive OCR Engine
- The Error Profiler is integrated with the Adaptive OCR Engine and ABBYY FineReader Engine in order to improve the suspicious flag (only if relevant improvements can be demonstrated from the evaluation)

June 2011 (M42)

- Final evaluation of overall integrated workflow finished, indicating any remaining interoperability issues and/or room for additional interaction or tweaking of performance within the remaining timeframe

December 2011 (M48)

- A number of integrated IMPACT workflows are available that enable dictionary support for all languages that have been tackled in the course of the project

5. Conclusion

Since most of the technical (software) deliverables in IMPACT were only due at the end of 2009, apparently a lot of effort had to been taken in the first half of 2010 in order to provide a uniform and user-friendly technical framework that connects all the different strands of work and combines them in an overall integrated workflow for OCR processing. Progress up to date shows that the service-oriented architecture approach that has been chosen suits very well the technical requirements in terms of providing all the flexibility required to successfully integrate a vast number of IMPACT tools and prototypes developed on different platforms and even third party applications. Clear integration guidelines and direct communication lines between technical project management and individual software developers as well as stakeholders do support the common understanding of the projects aims and needs towards technical integration and provide a sound basis for the integration of yet outstanding technical deliverables.

While there has always been a focus on innovation rather than performance, initial tests indicate promising results for scaling up the architecture of the Interoperability Framework to the needs of mass-digitisation. Through the Taverna system and the integrated website client, a user friendly way of interacting with IMPACT tools and applications as well as an example implementation of a fully web-based solution have been established and are ready to be demonstrated. In addition, the architecture exposes a number of beneficial characteristics such as for example the modular approach for workflow development which greatly supports evaluation tasks and capacity building in content holding institutions. By refocusing on two major XML standards for data exchange between different technical modules and integrating the evaluation tools with the Interoperability Framework, IMPACT not only creates a means for document processing on a large scale, but also for quality checking the results of OCR workflows against ground truth.

While most of the interoperability issues that have been identified alongside meetings in Innsbruck in February 2010 and Paris in April 2010 have been resolved by now, a number of challenges still remain. The delay that has been caused by the lack of ground truth and the complexities encountered in the production of huge numbers of ground truth for the demonstrators has so far prevented further quantitative evaluation of the Interoperability Framework and IMPACT tools and applications using a larger number of documents. This is expected to be mitigated in the course of the second half of 2010 when approximately 50.000 ground truth files will become available to the project and allow for an exact assessment of the increase in OCR accuracy that IMPACT solutions in terms of throughput and performance with regard to particular source material.

Another task that still requires attention lies in finding the optimal implementation strategy for integrating the outcomes of language work in the project with the overall OCR process and making them accessible in a convenient way.

Finally, the question of how technical outcomes from IMPACT will become available to a wider audience after project end and how they will be sustained through the Centre of Competence requires careful licensing and packaging of the technical framework and its components during the final one and a half years of the project.