D2.5

LR Switchboard (software)

Document information

Title	LR Switchboard (software)
ID	CLARINPLUS-D2.5 (CE-2016-0881)
Author(s)	Claus Zinn
Responsible WP leader	Dieter Van Uytvanck
Contractual Delivery Date	2016-10-31
Actual Delivery Date	2016-10-31
Distribution	Public
Document status in workplan	Deliverable (software)

Project information

Project name	CLARIN-PLUS
Project number	676529
Call	H2020-INFRADEV-1-2015-1
Duration	2015-09-01 – 2017-08-31
Website	www.clarin.eu
Contact address	contact-clarinplus@clarin.eu

Table of contents

1 Executive Summary

The CLARIN infrastructure gives users access to an increasingly rich set of language-related resources, using the Virtual Language Observatory, the Federated Content Search, and the Virtual Collection Registry. While there is ample support for searching resources using metadata-based search, or full-text search, or aggregating resources into virtual collections, there is little support for users to help them processing resources in one way or another. While there is a considerable number of processing software in the CLARIN world, there is no single point of access where users can find tools to fit their needs and the resource they have. In this deliverable, we present the CLARIN Language Resource Switchboard (LRS), which aims at helping users to connect resources with the tools that can process them. The LRS lists all applicable tools for a given resource, lists the tasks the tools can achieve, and invokes the selected tool in such a way so that processing can start immediately without any or little prior tool parameterization.

The deliverable reports on Task 2.3.1 of the CLARIN-PLUS project.

2 Introduction

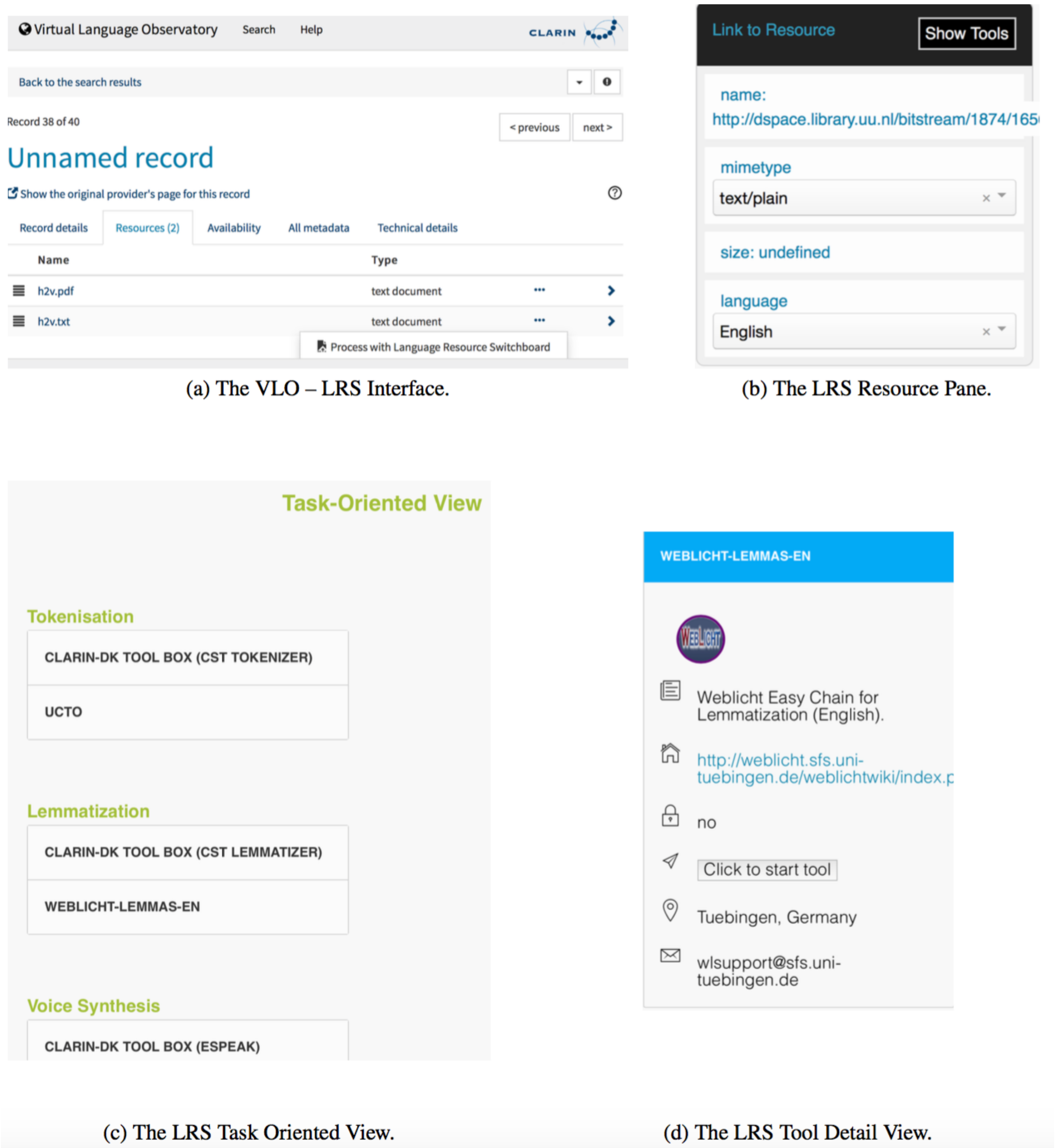
The pan-European CLARIN project is an e-Humanities project that is building an e-science infrastructure for language-related resources. Among the pillars of the infrastructure are the *Virtual Language Observatory* (VLO) that gives users a metadata-based access to language resources [1], the *Federated Content Search* (FCS) that gives users a full-text search across resources [2], and the *Virtual Collection Registry* (VCR), where users can collect resources in a virtual set [3]. CLARIN makes use of the *Component MetaData infrastructure* (CMDI) to describe resources in a common, flexible language. Persistent identifiers based on the Handle system (https://www.handle.net) ensure a persistent URL addressing of resources.

In the CLARIN world, there also is an increasing number of tools available to process language-related resources in a manifold manner. While a part of the tools must be installed locally on users' desktop machines, there also exists a good number of browser-based tools and web services.

The CLARIN Language Resource Switchboard (LRS) aims at bridging the gap between resources (as identified in the VLO, FCS, and VCR) and tools that can process these resources in one way or another. The LRS can be seen as a Virtual Tool Registry. For a given resource, it identifies all tools that can process the resource. It then sorts the tools in terms of the tasks they perform, and presents a task-oriented list to the user. Users can then select and invoke the tool of their choosing. By invoking the tool, all relevant information about the resource in question is passed onto the tool, and the tool opens with most information gathered by the switchboard. This makes it easy for users to identify the right tools for their resource, but also to use the chosen tool in the most effective way possible.

3 The LRS in Action

Consider the scenario where a researcher uses the VLO to find an English text that she then would like to investigate further. On the VLO search results page, the user can now click on the · · · area to invoke the LRS with this resource, see Fig. 1(a). In a new browser tab, the LRS opens and shows a resource pane that depicts all relevant information about the resource, see Fig. 1(b). The user is free to correct this metadata, before clicking on 'Show Tools' to get to the task-oriented view, shown in Fig. 1(c). If the user is interested in, say, the lemmatization task, she may wish to get more information about the two tools offered, in which case more detailed information about the chosen tool is given, see Fig. 1(d). When the user then clicks on 'Click to start tool', the chosen tool, here WebLicht, opens in a new browser tab. WebLicht obtains from the LRS a reference to the resource, the resource's mimetype and language as well as the chosen task. WebLicht opens with the predefined easy chain for lemmatization, loads itself the resource, and sets all relevant parameters so that the user is left to click on WebLicht's RUN command to start the processing chain. No further user action is required to parameterize WebLicht for this.



Unnamed record

Name	Type
h2v.pdf	text document
h2v.txt	text document

name: http://dspace.library.uu.nl/bitstream/1874/165

mimetype: text/plain

size: undefined

language: English

Tokenisation
- CLARIN-DK TOOL BOX (CST TOKENIZER)
- UCTO

Lemmatization
- CLARIN-DK TOOL BOX (CST LEMMATIZER)
- WEBLICHT-LEMMAS-EN

Voice Synthesis
- CLARIN-DK TOOL BOX (ESPEAK)

WEBLICHT-LEMMAS-EN

Weblicht Easy Chain for Lemmatization (English).

http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.p

no

Tuebingen, Germany

wlsupport@sfs.uni-tuebingen.de

(a) The VLO – LRS Interface.

(b) The LRS Resource Pane.

(c) The LRS Task Oriented View.

(d) The LRS Tool Detail View.

Figure 1. The LRS in Action.

4 Specification and Architecture

The initial specification of the LRS Switchboard is given in the CLARIN PLUS Report CE-2015-0684, which is available at https://office.clarin.eu/v/CE-2015-0684-LR_switchboard_spec.pdf

Fig. 2 depicts the architecture of the switchboard. Its back-end consists of three main components: a *profiler*, an *app registry*, and a *matcher*.

The **Resource Profiler** is responsible for identifying those resource characteristics for which applicable tools can be listed. The current version of the LRS takes into account the resource's mimetype and language, whose values are transferred from the VLO to the LRS. The profiler then makes use of the REST-based services of the Apache TIKA software to double-check the given mimetype and language information, see http://tika.apache.org .

In a future version, the LRS profiler may take into account the entire CMDI-based description for a resource, but also the users' VLO browser history; the facets that users selected to identify some resource encode, at least to some extent, the user's search intent.

The **App Registry** manages a set of metadata descriptions, with each element describing a tool configuration for the LRS. Fig. 3 depicts the metadata entry for the CLARIN-DK toolbox, offering a named entity analysis. For the CLARIN user, relevant metadata include the title of the application, an English description about its capabilities, and contact information about the tool provider. For the LRS, the relevant parts are the tool's task description (using a controlled vocabulary, e.g., "tokenization", "part-of-speech-tagging", "optical character recognition"), an ISO 639-3 based identification of the languages the tool can handle, and the mimetypes it can process. For tool invocation, the metadata holds the tool's web address and a list of parameters the tool understands. With this information, a URL can be constructed where all relevant information is URL-encoded.

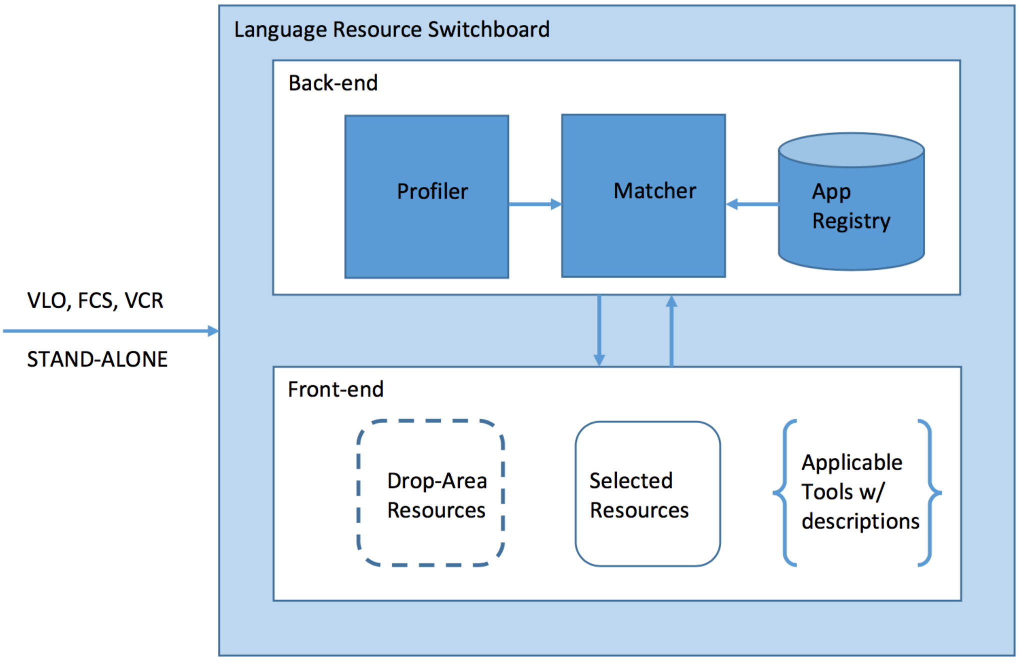
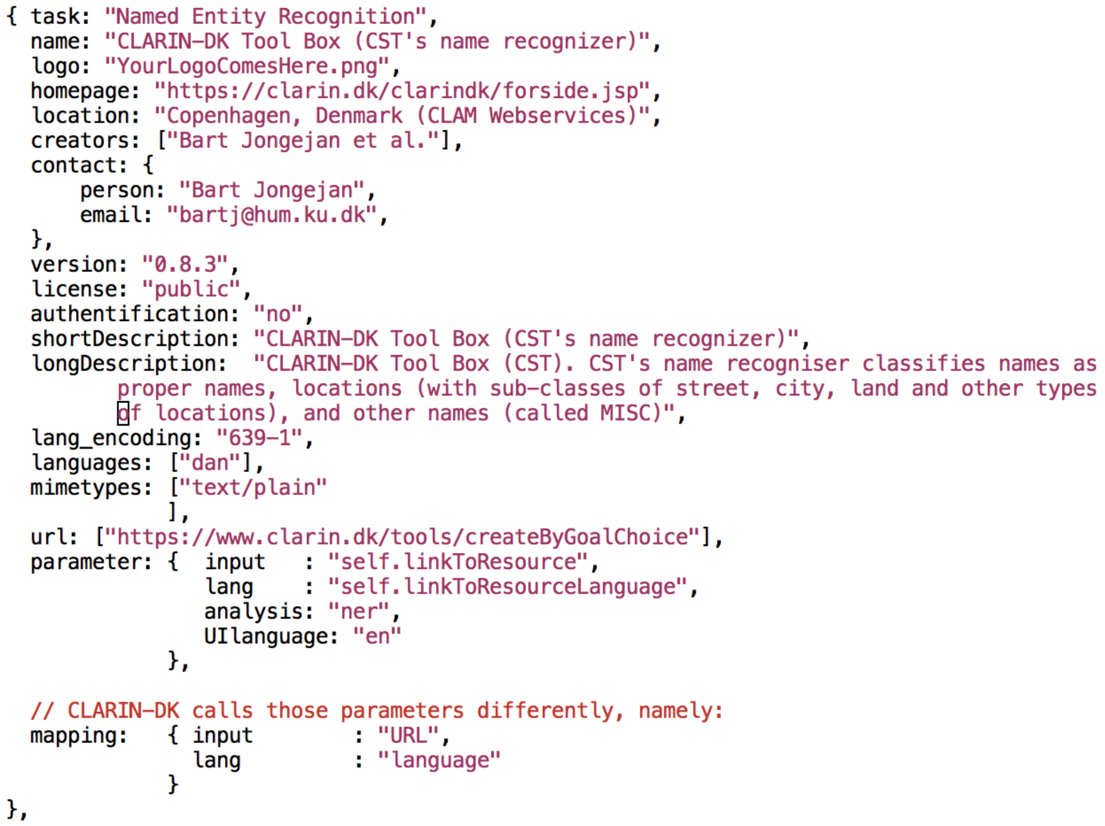


Figure 2. The LRS Architecture.

The **Matcher** uses the resource's metadata from the profiler, and the tools' metadata from the application registry to find matches. For the given resource profile, it computes a list of all applicable tools and the analyses they offer. For the time being, only the resource' mimetype and language are taken into account. A future version of the LRS might complement the tool profile with a user profile that holds information about a user's access rights. Here, the user can ask the switchboard to only process resources and only list tools that the user has access to.

The **LRS User Interface** can be invoked from the VLO (see Fig.1(a)), but there is also a LRS stand- alone version [4]. Here, users can upload their resources via a simple file drag and drop mechanism. The LRS then determines the resource's mimetype and language (using Apache TIKA), as shown in Fig.1(b), which the users can correct if necessary. In the future, the LRS may also offer a REST-based interface where a JSON structure of applicable tools will be returned given the resource or its metadata is being provided.



```
{ task: "Named Entity Recognition",
  name: "CLARIN-DK Tool Box (CST's name recognizer)",
  logo: "YourLogoComesHere.png",
  homepage: "https://clarin.dk/clarindk/forside.jsp",
  location: "Copenhagen, Denmark (CLAM Webservices)",
  creators: ["Bart Jongejan et al."],
  contact: {
      person: "Bart Jongejan",
      email: "bartj@hum.ku.dk",
  },
  version: "0.8.3",
  license: "public",
  authentification: "no",
  shortDescription: "CLARIN-DK Tool Box (CST's name recognizer)",
  longDescription:  "CLARIN-DK Tool Box (CST). CST's name recogniser classifies names as
         proper names, locations (with sub-classes of street, city, land and other types
         of locations), and other names (called MISC)",
  lang_encoding: "639-1",
  languages: ["dan"],
  mimetypes: ["text/plain"
             ],
  url: ["https://www.clarin.dk/tools/createByGoalChoice"],
  parameter: {  input   : "self.linkToResource",
                lang    : "self.linkToResourceLanguage",
                analysis: "ner",
                UIlanguage: "en"
             },

  // CLARIN-DK calls those parameters differently, namely:
  mapping:   { input        : "URL",
               lang         : "language"
             }
},
```

Figure 3. The metadata entry for the CLARIN-DK toolbox.

5 Implementation and Status

5.1 Software Implementation Details.

The LRS Switchboard is based on the following technologies (version numbers correct at the time of writing).

- React, a Javascript library for building user interfaces, see https://facebook.github.io/react/. Version used at time of writing: v.15.3.2
- Nodejs version 5.12.0 (see https://nodejs.org/en/), together with the Javascript package manager npm (v3.8.6), see https://www.npmjs.com .
- A number of node-based libraries, including the altjs package, see http://alt.js.org, and the alt container package, see http://alt.js.org/docs/components/altContainer/ .
- Webpack, a Javascript module bundler, see https://webpack.github.io .
- Apache Tika for analysing language resources with respect to the mimetype and language, see https://tika.apache.org.
- JSON (JavaScript Object Notation) is a lightweight data-interchange format, see http://www.json.org .

A number of other Javascript libraries is also been used. The webpack configuration file lists all dependencies. The source code is being maintained in a GitHub repository, see https://github.com/clarin-eric/LRSwitchboard . The command 'npm build' compiles all sources into a build directory; all files of this build directory need to be copied to the web server that serves the LR switchboard.
At the time of writing, the server address https://www.clarin.eu/switchboard is being used, pointing to a server hosted at the University of Tübingen, running on Ubuntu, release 14.04.4 LTS. In the future, we might want to complement the short URL https://vlo.clarin.eu with https://switchboard.clarin.eu to point to the standalone version of the LRS.

At the time of writing, the LRS makes use of a special edition of WebLicht that **bypasses** CLARIN authentication via Shibboleth.

The LR Switchboard has been connected to the VLO. For this a URL parameter passing API has been designed to call the LRS with all required information.

The LRS standalone version. In deviation to the project proposal, a standalone version of the LRS has also been developed. In this version, users can just upload their resources to the LRS, to then obtain a list of all applicable tools, sorted by the tasks they promise to achieve.

5.2 Applications that have been registered with the LRS

At the time of writing (mid-October 2016), a total of 35 applications and 11 web services have been connected to the switchboard. The first table gives an overview for all browser-based software.

Name	Task	Language(s)	Mime-types
Ucto	Tokenisation	nld, eng, deu, fra, ita, fry	text/plain
CST Tokenizer (CLARIN-DK Tool Box)	Tokenisation	bul, ces, dan, deu, ell, eng, spa, est, fas, fra, hun, isl, ita, lat, mkd, nld, pol, por, ron, rus, slk, slv, srp, tur, ukr	many formats
CST Lemmatizer (CLARIN-DK Tool Box)	Lemmatization	see above	many formats
CST Name Recognizer (CLARIN-DK Tool Box)	Named Entity Recognition	dan	many formats
espeak (CLARIN-DK Tool Box)	Voice Synthesis	many languages	many formats
Weblicht-Lemmas-EN	Lemmatization	eng	text/plain, text/rdf, application/pdf, application/msword, application/vnd.openx

			mlformats-officedocument.wordprocessingml.document
Weblicht-Tokenization-TK	Tokenisation	tur	see above
Weblicht-Lemmas-DE	Lemmatization	deu	see above
Weblicht-POSTags-Lemmas-DE	PoS Tagging	deu	see above
Weblicht-POSTags-Lemmas-IT	PoS Tagging	ita	see above
Weblicht-POSTags-Lemmas-EN	PoS Tagging	eng	see above
Weblicht-Dep-Parsing-NL	Dependency Parsing	nld	see above
Weblicht-Dep-Parsing-EN	Dependency Parsing	eng	see above
Weblicht-Morphology-DE	Morphology Analysis	deu	see above
Weblicht-Morphology-EN	Morphology Analysis	eng	see above
Weblicht-NamedEntities-DE	Named Entity Recognition	deu	see above
Weblicht-NamedEntities-EN	Named Entity Recognition	eng	text/plain
Tesseract | CuneiForm (CLARIN-DK Tool Box)	OCR Engine	dan, eng, generic	image/gif, image/jpeg, image/png, image/tiff
Voyant Tools	Text Analytics	eng, deu, spa, nld, fra, generic	text/plain, test/rdf, application/pdf, application/doc, others
T-scan	Text Analytics	nld	text/plain
Oersetter (NLD-FRY)	Machine Translation	nld	text/plain
Oersetter (FRY-NLD)	Machine Translation	fry	text/plain
Fowlt	Spelling correction	eng	text/plain
Fowlt (xml+folia)	Spelling correction	eng	text/folia+xml
Frog	NLP suite for Dutch	nld	text/plain
Frog (folia+xml)		nld	text/folia+xml
"FoLiA-stat	N-Gramming	nld, generic	text/folia+xml
Valkuil	Spelling correction	nld	text/plain
Valkuil (folia+xml)	Spelling correction	nld	text/folia+xml
Colibri Core	N-Gramming	nld, eng, deu, fre, esp, por, fry, generic	text/plain

Colibri Core (folia+xml)	N-Gramming	nld, eng, deu, fre, esp, por, fry, generic	text/folia+xml
Alpino	Dependency Parsing	nld	text/plain

The following table gives an overview for all **web services** that have been connected to the switchboard:

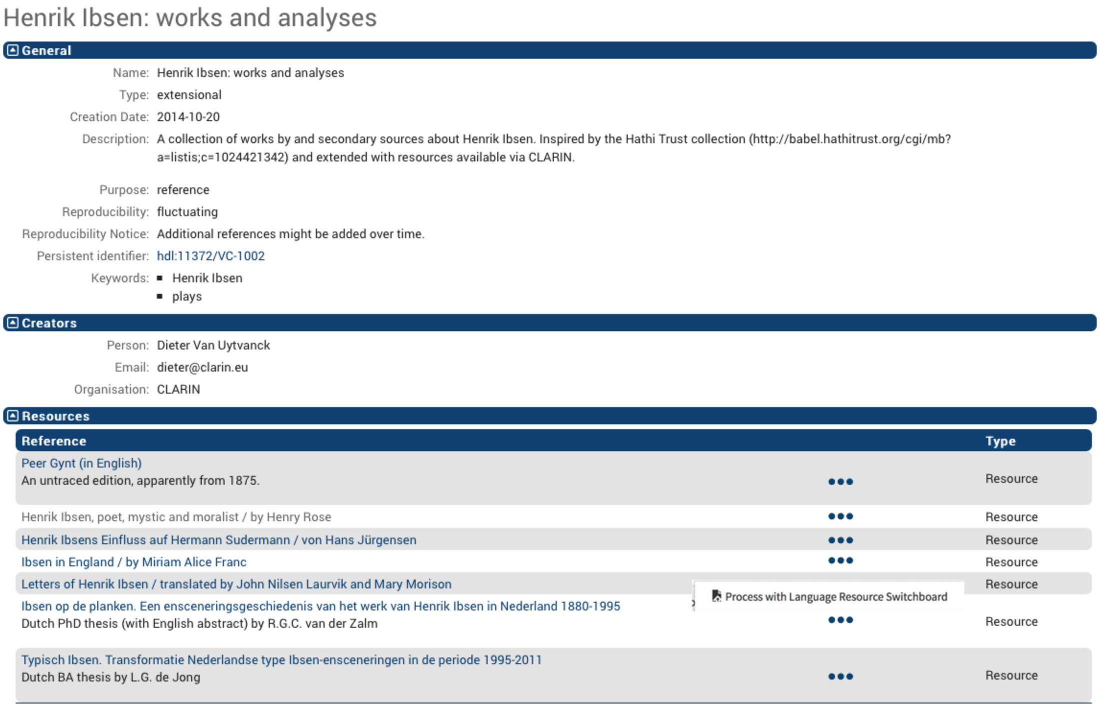
Mary TTS	Voice Synthesis	deu	text/plain
runMinni	Phonetic Transcription	deu	audio/wav
OxGarage	Conversion	many languages	application/tei+xml, text/tei+xml;format-variant=tei-dt
TEI↔TCF encoder+decoder	Conversion	generic (metadata)	application/tei+xml, text/tei+xml;format-variant=tei-dta, application/xml;format-variant=weblicht-tcf
CMDI2DC	Metadata Format Conversion	generic (metadata)	text/xml
DC2MARC21	Metadata Format Conversion	generic (metadata)	text/xml
MARC2EAD	Metadata Format Conversion	generic (metadata)	text/xml
Marc2MODS	Metadata Format Conversion	generic (metadata)	text/xml
Marc2RDFDC	Metadata Format Conversion	generic (metadata)	text/xml
MODS2RDF	Metadata Format Conversion	generic (metadata)	text/xml
KER	Keyword Extractor	ces, eng	text/plain
NameTag	Named Entity Recognition	ces, eng	text/plain

6 Integration with other CLARIN Services

The CLARIN-PLUS work-plan also foresees the integration of the switchboard with the CLARIN Virtual Collection Registry and the CLARIN Federated Content Search.

6.1 Integration with the Virtual Collection Registry

Fig. 4 shows a screenshot of the Virtual Collection Registry, displaying a collection of work by Henrik Ibsen. The collection consists of seven resources, see the reference section in the bottom half of the figure. Each reference has a title, a type (either “Resource” or “Metadata”), a link to the resource (preferably, a persistent identifier pointing to the resource), and optionally, a description. Note that there is no CMDI-based metadata attached to a reference, such as, for instance, the resource's mimetype or language.



Henrik Ibsen: works and analyses

General

Name: Henrik Ibsen: works and analyses

Type: extensional

Creation Date: 2014-10-20

Description: A collection of works by and secondary sources about Henrik Ibsen. Inspired by the Hathi Trust collection (http://babel.hathitrust.org/cgi/mb?a=listis;c=1024421342) and extended with resources available via CLARIN.

Purpose: reference

Reproducibility: fluctuating

Reproducibility Notice: Additional references might be added over time.

Persistent identifier: hdl:11372/VC-1002

Keywords:
- Henrik Ibsen
- plays

Creators

Person: Dieter Van Uytvanck

Email: dieter@clarin.eu

Organisation: CLARIN

Resources

Reference		Type
Peer Gynt (in English) An untraced edition, apparently from 1875.	●●●	Resource
Henrik Ibsen, poet, mystic and moralist / by Henry Rose	●●●	Resource
Henrik Ibsens Einfluss auf Hermann Sudermann / von Hans Jürgensen	●●●	Resource
Ibsen in England / by Miriam Alice Franc	●●●	Resource
Letters of Henrik Ibsen / translated by John Nilsen Laurvik and Mary Morison		Resource
Ibsen op de planken. Een ensceneringsgeschiedenis van het werk van Henrik Ibsen in Nederland 1880-1995 Dutch PhD thesis (with English abstract) by R.G.C. van der Zalm	●●●	Resource
Typisch Ibsen. Transformatie Nederlandse type Ibsen-ensceneringen in de periode 1995-2011 Dutch BA thesis by L.G. de Jong	●●●	Resource

Figure 4: From the Virtual Collection Registry to the Switchboard.

We manipulated the screenshot to show a number of dots associated with each resource. When users click on them, a pop-up window "Process with Language Resource Switchboard" opens, similar to the VLO behaviour depicted Fig. 1(a). When the user clicks on the pop-up, the link associated with the resource is sent to the switchboard for subsequent processing. Given the lack of resource metadata on the VCR side, it is the switchboard's task to identify the resource's language and mimetype before suggesting applicable tools that can process the resource.

In the future, the following example link will invoke the switchboard from the VCR side:

http://weblicht.sfs.uni-tuebingen.de/clrs/#/vcr/http:%2F%2Fhdl.handle.net%2F10932%2F00-01B8-AF59-4FB9-9201-B%09

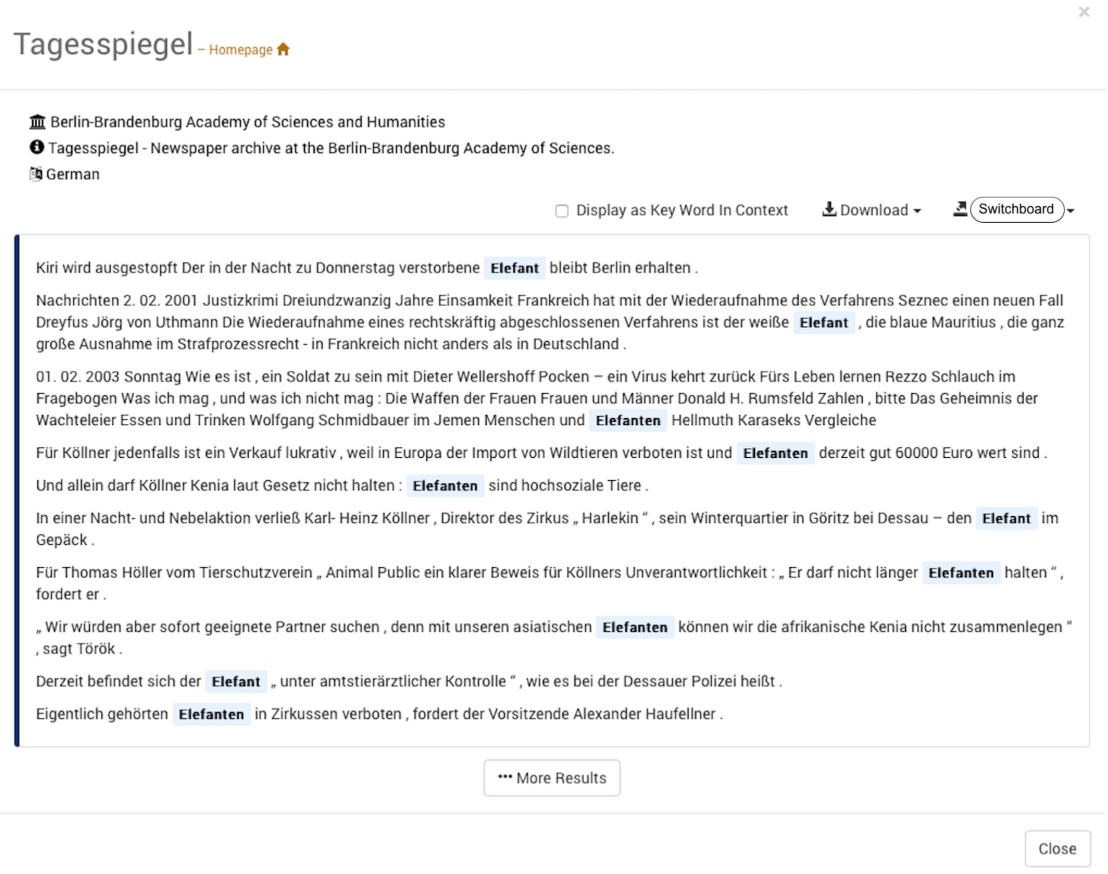
In contrast to the invocation of the switchboard from the VLO, only the reference to the resource is given (and #/vlo/ is replaced by #/vcr/).

6.2 Integration with Federated Content Search

Fig. 5 depicts a screenshot of CLARIN's Federated Content Search, which we have also manipulated to illustrate its integration with the switchboard. For this, note the switchboard icon at the top right-hand side. Once it is clicked it will forward the resource to the switchboard by invoking the following URL, for instance:

http://weblicht.sfs.uni-tuebingen.de/clrs/#/fcs/http:%2F%2Fhdl.handle.net%2F10932%2F00-01B8-AF59-4FB9-9201-B%09

Similar to the VCR case, there is a little metadata available, so that it is the switchboard's task to identify the resource's mimetype and language.



Berlin-Brandenburg Academy of Sciences and Humanities

Tagesspiegel - Newspaper archive at the Berlin-Brandenburg Academy of Sciences.

German

Kiri wird ausgestopft Der in der Nacht zu Donnerstag verstorbene **Elefant** bleibt Berlin erhalten .

Nachrichten 2. 02. 2001 Justizkrimi Dreiundzwanzig Jahre Einsamkeit Frankreich hat mit der Wiederaufnahme des Verfahrens Seznec einen neuen Fall Dreyfus Jörg von Uthmann Die Wiederaufnahme eines rechtskräftig abgeschlossenen Verfahrens ist der weiße **Elefant** , die blaue Mauritius , die ganz große Ausnahme im Strafprozessrecht - in Frankreich nicht anders als in Deutschland .

01. 02. 2003 Sonntag Wie es ist , ein Soldat zu sein mit Dieter Wellershoff Pocken – ein Virus kehrt zurück Fürs Leben lernen Rezzo Schlauch im Fragebogen Was ich mag , und was ich nicht mag : Die Waffen der Frauen Frauen und Männer Donald H. Rumsfeld Zahlen , bitte Das Geheimnis der Wachteleier Essen und Trinken Wolfgang Schmidbauer im Jemen Menschen und **Elefanten** Hellmuth Karaseks Vergleiche

Für Köllner jedenfalls ist ein Verkauf lukrativ , weil in Europa der Import von Wildtieren verboten ist und **Elefanten** derzeit gut 60000 Euro wert sind .

Und allein darf Köllner Kenia laut Gesetz nicht halten : **Elefanten** sind hochsoziale Tiere .

In einer Nacht- und Nebelaktion verließ Karl- Heinz Köllner , Direktor des Zirkus „ Harlekin " , sein Winterquartier in Göritz bei Dessau – den **Elefant** im Gepäck .

Für Thomas Höller vom Tierschutzverein „ Animal Public ein klarer Beweis für Köllners Unverantwortlichkeit : „ Er darf nicht länger **Elefanten** halten " , fordert er .

„ Wir würden aber sofort geeignete Partner suchen , denn mit unseren asiatischen **Elefanten** können wir die afrikanische Kenia nicht zusammenlegen " , sagt Török .

Derzeit befindet sich der **Elefant** „ unter amtstierärztlicher Kontrolle " , wie es bei der Dessauer Polizei heißt .

Eigentlich gehörten **Elefanten** in Zirkussen verboten , fordert der Vorsitzende Alexander Haufellner .

Figure 5: From the Federated Content Search to the Switchboard.

7 Conclusion and Outlook

Two big issues that affect the usability of the LR switchboard are the quality of the metadata in the VLO, and access restrictions to resources and tools. The VLO lists a considerable number of resources with incorrect or incomplete mimetypes. Searching, for instance, for resources of type "text/plain" might yield textual resources that are not plain, but enriched with other annotations. When users load such a resource in the LRS, and subsequently process the resource with a tool of their choice, they may find out that the resource's content is not what they expected. Similarly, accessibility and authentication issues affect the usability of the LR switchboard. When the user invokes an applicable tool in the switchboard, the tool might not be able to download the resource from the resource provider. Also, some tools require authentication, and some users may not have the proper access rights to make use of the tools. For this, a number of user delegation issues need to be tackled, with many technical intricacies involved [7]. Also, public resources should be marked accordingly in the resource metadata to prevent cases where users have access to the tools but not to the resources, or *vice versa*.

The switchboard also faces the risk that tool makers fail to make an effort of getting their tools connected to the switchboard. To address this risk, we have attempted to minimise the implementation work required. The browser-based tools must be adapted to parse and interpret parameters added the tools' URL invocation path. For the time being, one parameter is crucial, namely, for encoding the resource's URL to be processed. A tool connected to the LRS should be able to extract the parameter from the tool's invocation URL, and fetch the resource from the given location. Many tools already provide such functionality, and the switchboard "knows" the tools' parameter name so that the tool is called accordingly (see the mapping slot in the tools' metadata description). We hope that the switchboard's attractiveness (which we increased by the addition of web services) convinces tool developers to make the necessary changes.

In the near future, the LRS will need to address a number of issues:

Interoperability Aspects. At the time of writing, the tools that feature in the LRS can hardly be used together, or chained one after the other. However, the LRS was never designed to be a workflow engine such as WebLicht. The task of the LRS is rather simple: given some information about a resource, help users to find and start a tool that can process the resource in one way or another. Once the user has been directed to the tool (the chosen tool is started in a new browser tab), LRS's engagement ends; LRS is unaware of the processing, and whether the processing succeeded or failed. Of course, users may manually save the output of the tool to a file, and then upload it to the LRS to find post-processing tools. But such tools are rarely available as most tools have their proprietary output format that other tools cannot read. In this respect, note that the integration of tools into the WebLicht workflow engine is very labour-intensive as tools must be adapted to become capable of reading and writing TCF conform data. Tools that were connected to the LRS required relatively little adaptation; they were modified so that tools were capable of processing a number of input parameters - no data formats were changed.

Management Aspects. So far, the process of adding a tool to the LRS is not mechanised. In the future, the LRS may offer an extra web form, where tool owners can register their tool with the provision of metadata (see Fig. 3), or where they can update such data. In an update step, tool owners may advertise a new version of their tool, an alternative web address where it can be reached, or even an alternative mirroring site. We anticipate the need for such tool-supported management software once the LRS is more widely known and used.

Legal Aspects. In the LRS stand-alone version, an uploaded resource is temporarily stored at the Max Planck Computing and Data Facility. When the user chooses a tool to process the resource, the tool downloads the resource from there. Resources are hence passed back and forth to different servers, and at the time of writing, legal aspects have not been taken into account. Such aspects need to be addressed in due time, in particular, for resources that do not fall under open access policies. Note that this issue relates to the authentication and authorization aspect mentioned earlier.

Scalability Aspects. The processing of large files is problematic as some tools may fail to process them in a timely manner. WebLicht, for instance, imposes a size limit of 3 megabytes (independent of the chosen workflow), but other tools are less explicit about this. Moreover, the size of a resource is sometimes not visible in the VLO; in this case, the size is only detectable once the resource has been downloaded to a hard disk or to the temporary file storage server. In any case, the LRS is currently not using file size information, and hence will not hide tools that may face problems with large files. This issue will need to be addressed.

At the time of writing, the LRS, and hence its associated tools, offer no ``batch mode'', where users can pack (zip) files of a common type together to have them processed in sequential order. For this task, users should directly use services amenable to such processing such as *WebLicht as a Service*, see [5].

Some tools need more than a single file to offer their services. The WebMAUS Basic software, *e.g.,* requires both a waveform audio file and a plain text file for phonetic segmentation [6]. The stand-alone LRS could address the issue of processing file pairs or triples by adapting the upload mechanism in the graphical user interface accordingly. The solution is less clear when looking from the VLO side. The current solution only allows a single resource to be forwarded to the switchboard, see Fig. 1(a). In the long term, a 'shopping basket' metaphor could be considered where VLO users can add multiple resources to a switchboard basket, before asking for a transferal to the switchboard. Before the transferal, the VLO would pack the basket's content into a bundle, attach metadata to the bundle based on its content, and pass the bundle to the switchboard, which in turn, must be adapted to handle the bundle accordingly.

Note that the VLO already hosts many metadata records that describe a 'bundle' or 'session', linking together multiple related resources. Consequently, the simple URL encoding of information for invoking the switchboard from the VLO may already require adaption to cope with such complexity. Also, similar considerations may apply for the interface between the Virtual Collection Registry and the switchboard.

Usability Testing. As of October 2016, the LRS has only been exposed to a limited number of users. While the feedback received has been overwhelmingly positive, a more systematic user testing must take place, for both the standalone version and the LRS version that is invoked via the CLARIN VLO (and later, the CLARIN FCS and the CLARIN VCR). The user testing should test the LRS with respect to cross-browser compatibility.

Integration of the Switchboard with the FCS and the VCR. At the time of writing, the switchboard has only been connected to the Virtual Language Observatory. In the future, the switchboard shall also be easily accessible to process resources from the Virtual Collection Registry and the Federated Content Search. The usage scenarios we have discussed in Sect. 6 are a good starting point for such integration.

Minor issues. There are a number of potential minor code changes to improve the usability of the LRS. We list them as a brief to-do list:

- Allow users to upload a pair of files, see WebMaus scenario above.
- Restrict the number of mimetypes and languages the users can choose from; these should be automatically computed from the entries of the app registry.
- The mimetype specificity has to be tackled and harmonized across the CLARIN community, *e.g.,* have "text/cmdi+xml" and "application/tei+xml" rather than "text/xml".
- Add a switch to enable immediate tool start for all tools that offer this feature.
- Offer a faceted browser experience for the standalone version where all tools are classified in terms of a small number of facets ("language", "mimetype", "software-type" *etc.*), that is, offer some kind of a "Virtual Tool Observatory" without the need to upload a resource.
- Consider a feedback switch where users can "mark" the tools they have used via the LRS (satisfaction with tool's performance, results *etc.*)

- Consider a user login to the LRS once user delegation is being sorted out. Guest users may not have access to some tools, users authenticated via Shibboleth have access to all tools and resources with that "clearance".
- Consider a REST-based interface to the switchboard.

Outlook. The LR switchboard has received positive feedback during the last CLARIN centre meeting in Utrecht and the CLARIN 2016 Annual Conference in Aix-en-Provence. Participants liked the stand-alone version of the LRS, and asked for the version to remain available once the LR switchboard is integrated with the VLO, FCS, and the VCR.

During the CLARIN 2016 conference, participants also asked for a number of additional features to be added to the LRS, and developers elicited help to get their tools integrated with the switchboard. One participant addressed the metadata required for tool registration, and asked whether all data could be taken automatically from the CMDI-based metadata that a tool might already have. This relates to the management aspects discussed earlier; a future version of the LRS might offer a tool management component where existing, CMDI-based tool metadata is taken into account.

During the conference, it was also discussed whether the interaction between the LRS and the tools should become a two-way communication. Once the LRS has invoked the tool with a URL-encoded command, the tool could respond with a status message that informs the switchboard about the processing status, *e.g.,* "task succeeded", "task failed". The switchboard could then use the tools' responses to maintain statistical data about tool use (for tools started via the switchboard). One participant wished that users were able to rate tools via the switchboard to gather feedback data, but a two-way communication between the switchboard and the tools might yield a more consistent data stream, while at the same time keeps tool developers in control. A future version of the LRS might offer a protocol that defines a two-way communication between the switchboard and associated tools.

During the conference, participants also asked whether the switchboard could act as a format harmonization device, making it capable of intelligently performing the necessary format conversions so that tools become applicable and potentially chainable. The switchboard would thus extend its mediation service. A future version of the LRS might give users more detailed help about the various input and output formats that tools require and produce. Here, the switchboard might also suggest possible format conversions, given that users cannot identify a suitable tool to process their resources.

During the conference, participants also asked whether the switchboard could also list tools that cannot directly be invoked by the switchboard, either because a given tool has not been integrated yet with the switchboard, or because the tool is desktop-based software and must be installed on users' machines. A future version of the switchboard might offer a section "Other tools" that covers such cases.

From the vivid discussions, it showed that the LR switchboard is indeed perceived as the missing link between language-related resources and the tools that can process them. The stand-alone version allows users to bypass some of the issues discussed above, and also to play with the switchboard with users "personal" linguistic resources.

References

[1] The CLARIN Virtual Language Observatory,
see https://vlo.clarin.eu/ .

[2] The CLARIN Federated Content Search,
see https://www.clarin.eu/contentsearch .

[3] The CLARIN Virtual Collection Registry,
see https://www.clarin.eu/vcr .

[4] The CLARIN Language Resource Switchboard,
see https://www.clarin.eu/switchboard .

[5] WebLicht as a Service, see https://weblicht.sfs.uni-tuebingen.de/WaaS/

[6] WebMaus Basic, see https://clarin.phonetik.uni-muenchen.de/BASWebServices/ -/services/WebMAUSBasic.

[7] J. Blumtritt, W. Elbers, T. Goosen, M. Hinrichs, W. Qiu, M. Salle, M. Windhouwer. *User Delegation in the CLARIN Infrastructure.* CLARIN 2014 Selected Papers; Linkoping Electronic Conference Proceedings, 2014.

[8] Claus Zinn. The CLARIN Language Resource Switchboard. CLARIN 2016 Annual Conference, Aix-en-Provence, France, 2016.

[9] The Switchboard specification (Milestone 2.2), see https://office.clarin.eu/v/CE-2015-0684-LR_switchboard_spec.pdf