D2.3

Newly integrated webservices

Document information

Title	Newly integrated web services
ID	CLARINPLUS-D2.3 (CE-2016-0840)
Author(s)	Claus Zinn, Marie Hinrichs
Responsible WP leader	Erhard Hinrichs
Contractual Delivery Date	2016-08-31
Actual Delivery Date	2016-08-31
Distribution	Public
Document status in workplan	Deliverable

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

Over the years, members of the CLARIN community have developed a rich set of software to process language-related resources. There is desktop software such as GATE offering a full suite of natural language processing tools, and software that do a specific task such as lemmatization or dependency parsing very well. Desktop software has a few disadvantages, though. Their installation on the users' local machine often requires technical expertise, and often depends on a particular operating system, programming language environment, and other factors. With the advent of web-based systems, the CLARIN community has also developed tools that can be used in a web browser. Often the tools' API is also exposed via a REST interface, so that their functionality is accessible by other software.

Workflow engines make use of these developments and offer an environment where Natural Language Processing (NLP) services can be accessed, and where the output of one service feeds the input that another NLP service can consume. In the CLARIN community, there are a number of workflow engines that users can access to perform a wide range of NLP tasks. For historical reasons, the workflow engines include services that focus on a relatively small number of languages (in particular, German, Dutch, and English); in part because the engines' developers first integrated web services they knew of and had access to.

However, there is available a relatively large number of NLP tools for other CLARIN languages such as French, Italian, Czech *etc*. To increase their use and accessibility, an orchestrated effort in required to integrate these tools into the existing workflow engines.

The goal of the CLARIN PLUS project's subtask 2.3.2 is to assist tool developers in the integration of their tools into one or more of the established workflow engines. In this deliverable, we describe how this task has been tackled. The actions reported in the deliverable are complemented by the efforts undertaken in subtask 2.3.1, where a relatively large number of NLP tools have also been integrated in the CLARIN Language Resource Switchboard.

2 Introduction

In the CLARIN community, there are now a few workflow engines progressing towards an increasingly mature and stable state. So far, however, the workflow engines cater for only a relatively small number of languages, such as German and English for the workflow engine WebLicht [1], and Dutch for the workflow engine TTNWW [6]. So far, language support was somewhat bound to the native languages of the workflow engine developers, and the other tools they knew of and worked with. There is available, though, a relatively high number of web services that can process other languages (such as Italian, French, Czech, Slovak), usually stemming from other or new CLARIN member countries. The objective is to integrate and to make available these services in the existing workflow engines, and hence, to also increase their visibility and use, and also, to make the workflow engines more attractive to a larger CLARIN community. An integration of web services into workflow engines helps increasing their usability and establishes a user experience that is consistent across tools and the languages they can process.

In this deliverable, we report on our work to achieve this objective. It is divided into three subtasks: the improvement of documentation and tutorial material that developers can use to make available or to adapt their web service for the workflow engines; the provision of tools that developers can use to describe their web services with metadata to facilitate the integration effort; and the active support of workflow engine developers to the developers of web services. A large part depends on a compatible data exchange format across web services so that they can easily be chained together to assemble processing pipelines needed by the community to analyse language-related resources.

The remainder of the document is structured as follows. First, we give an overview of the workflow engines currently in use in the CLARIN community. This is followed-up by a brief survey of web services that we know of. At the end of Section 2, we describe the necessary work for web services to be integrated into the workflow engines. In Section 3, we describe the documentation task, with Section 4 taking a closer look at the data interchange format required for the integration of web services into tool chains. In Section 5, we list the web services that we have integrated into WebLicht. And in Section 6, we conclude.

2.1 Existing Workflow Engines

Workflow engines make available tools that process language-related resources in one way or another. The opportunity to find and execute tools in sequence in a web browser greatly improves the accessibility and usability of the tools. Neither do users need to be concerned with the installation of various tools on their desktop machine, nor do they need to bother whether the output of one tool is “understood” by the tool responsible for a subsequent processing stage. Based on a common data format, and formal specifications for tools’ input and output behaviour, the workflows’ pipelining engine supports users in selecting and chaining tools together to achieve the task at hand.

There are some workflow engines that aim to give users a web-based access to a good number of individual web services.

2.1.1 WebLicht

WebLicht (short for: *Web-Based Linguistic Chaining Tool*) is a web-based service environment for the integration and use of language-related processing tools [1]. It has been originally developed as part of the German D-SPIN project, is widely used, and has been continuously maintained ever since. With WebLicht, users do not need to install any software on their own computers or to concern themselves with the technical details involved in building tool chains. With WebLicht, users can either use pre-defined workflows for common tasks, or used an "Advanced Mode" to define their own chains. For developers, WebLicht has been designed to ease the integration and use of distributed web services with standardized APIs. Developers are helped at contributing their own web services to WebLicht using extensive documentation, sample services to build upon, and tools for the metadata description of their web services, in particular, for their input and output parameters.

At the time of writing, WebLicht has integrated about 75 tools, often available with different parameterizations so that over 275 preconfigured tool instances are available. The large majority of services process German and English text. A full tool overview can be obtained from WebLicht at https://weblicht.sfs.uni-tuebingen.de/weblicht/ , see "View Tool List".

2.1.2 TTNWW

The workflow engine TTNWW integrates and makes available existing software components for the automatic processing of Dutch language (for text and speech) that have been developed in the STEVIN and CGN projects. The components are made available as web-services in a simplified workflow system that enables researchers without much technical background to use standard *workflow* recipes, see [6].

At the time of writing, according to Marc Kemps-Snijders, *"the TTNWW service is currently out of operation".* He continues: *"We are working on a new version with a different set up. Expected delivery date will be later this year, sometime during the autumn, depending upon our other projects' workload."*

As a consequence, any subtask 2.3.2 related activities with regard to TTNWW has been halted until further notice.

2.1.3 CLARIN-DK Tool box

The CLARIN-DK Tool box offers users the automated creation and execution of workflows. When a user uploads his or her input file(s), the toolbox computes and lists the workflows available to yield a given target format. Once a target format is selected, the workflow is executed, and an e-mail is sent to the user when results are available, see https://www.clarin.eu/content/web-services . The CLARIN DK toolbox supports a number of different formats, namely, pdf, rtf, doc, docx, odt, txt, and html.

Web services offered by the CLARIN-DK Tool box have been integrated in to the CLARIN Language Resource Switchboard.

2.2 Existing Web Services

In the CLARIN community, there are a number of well-known web services providers:

- The CLARIN-DK web services, see https://www.clarin.dk/tools/ .
- The LST web services portal, see https://webservices-lst.science.ru.nl .
- The BAS web services, see https://clarin.phonetik.uni-muenchen.de/BASWebServices/#/services
- The NLP Services of Leipzig University, see http://wortschatz.uni-leipzig.de/axis/servlet/ServiceOverviewServlet
- The NLP Services for Polish of Wroclaw University, see http://demo.clarin-pl.eu/demo.html

A full overview is given on the CLARIN website, see https://www.clarin.eu/content/web-services .

Tools of the first three sites were integrated via the *CLARIN Language Resource Switchboard* (see below).

2.3 Integration of Web Services into Workflow Engines

The integration of web services into workflow engines requires time and effort. With a given REST interface already in place, the larger part of the workload is the adaptation of the software's data structures for the encoding of its input and output.
For web services to be integrated into WebLicht, developers must enable their web services to process TCF[1]-compliant input, and to generate TCF-compliant output. Also, for WebLicht, web services must be "advertised" via a CMDI-based profile for the description of web services (for details, see below). With the web service adapted and its existence advertised via a known and harvestable CLARIN centre repository, WebLicht's orchestration engine can then offer the web service to users whenever an input resource of the given characteristics is available.

In the remainder of the document, we focus on the integration of web services with regard to WebLicht. We also list the number of web services that we have made available via the CLARIN Language Resource Switchboard.

[1] Text Corpus Format, see http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/The_TCF_Format

3 Documentation (WebLicht)

The WebLicht project offers a number of tutorials to help developers add their REST-based web services to the WebLicht workflow engine. WebLicht's main *Developer Manual* can be found at

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

The manual gives developers documentation about RESTStyle Web Services, WebLicht's chaining algorithm, and the TCF format. The documentation on these pages is very detailed and gives developers also example projects to kick-start their integration efforts. Given the good status of the WebLicht documentation, we have only added a checklist for developers that summarises the basic guidelines to be followed. The checklist is given below:

Checklist for the integration of an existing web service into WebLicht

You have an existing web service that can be accessed via CURL or WGET, and which has been tested thoroughly? Please check that each of the following items:

1. *Make sure your web service accepts TCF-compliant input.*
2. *Make sure your web service generates TCF-compliant output.*
3. *Install the web service on a machine accessible to the public.*
4. *Obtain a persistent identifier to refer to the URL address of the web service.*
5. *Use the Comet tool to create the CMDI-based metadata to describe the web service in question.*
6. *Enter the metadata into a CLARIN center repository so that it can be harvested from there.*
7. *Check whether WebLicht has harvested the metadata from the aforementioned repository.*
8. *Check whether your web service is working as expected in the WebLicht environment.*

When working through the checklist, stay in close contact with the WebLicht Development Team.

All CLARIN centres should have access to the Handle system for the persistent identification of their resource (item 4). Also a CLARIN centre will usually operate its centre repository, or will have access to a centre repository of a partner organisation (item 6).

Making an existing web service a registered WebLicht-compatible web service requires some effort, mostly in adapting an existing web service in terms of its I/O behaviour. The TCF-based format is the basis of the WebLicht orchestration algorithm, which we describe in Section 4.

4 Common Data Exchange Format of Web Services (WebLicht)

For best integration of a new web service to take part in WebLicht's tool-chaining, its input/output format should be compatible with the other WebLicht web services. For this reason, most WebLicht web services make use of TCF, an XML-based machine-readable exchange format. The TCF format is defined in the RELAX-NG schema definition language. The latest TCF version v0.4 can be obtained from http://clarin-d.de/en/tutorials#tcf .

TCF defines a set of annotation layers that goes hand in hand with the tools that generate them. The TCF v.04 schema defines the annotation layers: "text", "tokens", "sentences", "lemmas", "part-of-speech", "Constituent Parsing", "Dependency Parsing", "Morphology", "Named Entities", "References", the Lexical-semantic annotations "synonymy", "antonymy", "hyponymy", and "hyperonymy", "Matches", "Word Splittings", "Geographical locations", "Discourse connectives", "Phonetics", "Text structure", "Orthography", and "Text Source".

Web services may add annotation layers, but may not delete or make changes to existing ones.

Note that the "TextSource" element allows users to embed (or to refer to) any format into TCF. For more details, see
http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/The_TCF_Format .

The TCF format was designed to be maximally flexible with regard to the addition of annotation layers. The format can develop, thus, with the development of new tools and their needs, new layers can be added when required. To ensure backward compatibility with existing tools, no changes will be made to existing layers, though.

Developers can make use of the TCF binding library. It abstracts from the particular format of TCF and its parsing. The library binds linguistic data, represented by XML elements and attributes in TCF, to linguistic data represented by Java objects. The library interface allows developers to easily access the linguistic annotations in TCF and to easily add new linguistic annotations. With the library, developers are thus dealing with Java objects at an abstract level, and they do not need to be concerned with reading and writing XML documents at the syntax level.

5 Integration of new web services (WebLicht, Switchboard)

We have defined a number of so-called WebLicht *easy-chains* to make the integration of existing tools more visible and to support a wider range of European languages:

- **Dutch**: dependency parsing (Alpino).
- **Italian**: tree tagger.
- **French**: tree tagger.
- **Turkish**: tokenizer.
- **Czech**: a translation engine to Slovak.

More web services will be integrated at a later stage.

Note that some tools appear in more than one workflow engine. The Alpino Dependency Parser, for instance, has been included now in TTNWW and also WebLicht. It is also available in the CLARIN Language Resource Switchboard.

With the WebLicht workflow engine focusing on processing text files, we have abstained from integrating web services that process metadata, in particular, the conversion of CMDI to bibliographic metadata formats. A proper integration would have over-stretched the underlying intention of the TCF format.

However, we have managed to integrate the web services that convert between different metadata standards with the CLARIN Language Resource Switchboard.

- **DC2Marc**: converting between Dublin Core Simple [DC] and the bibliographic format MARC 21 [MARC], based on an existing XSL provided by the Library of Congress.
- **Marc2EAD**: converting between MARC 21 and EAD [EAD].
- **Marc2MODS**: converting between MARC 21 and MODS [MODS].
- **Marc2RDFDC**: converting between MARC 21 and RDF-based Dublin Core.
- **MODS2RDF**: converting between MODS and RDF.

The converters make use of XSL-based stylesheets developed by the Library of Congress, and hence, make available metadata conversions prominent in the library world to the CLARIN community.

Moreover, we have written XSL-based stylesheets to convert between CMDI-based metadata and the bibliographic standards Dublin Core and MARC 21:

- **CMDI2DC**: converting between CMDI-based metadata and Dublin Core.
- **CMDI2Marc**: converting between CMDI-based metadata and MARC 21.

Also, the following web services have been experimentally integrated with the CLARIN Language Resource Switchboard:

- *Mary* Text-To-Speech engine.
- *runMinni* web service for phonetic transcription.
- *KER* Keyword Extractor for Czech and English.
- *NameTag* for Named Entity Recognition for Czech.
- *OxGarage* web service to transform XML-based TEI documents to plain text.
- *TEI to TCF converter* to map the TCF format to the TEI format and *vice versa.*

From the CLARIN-DK portal, we have integrated the CST Tokenizer, the CST Lemmatizer, and the CST Named Entity Recognizer as well as eSpeak for voice synthesis, and Tesseract / Cuneiform for optical character recognition. These web services are accessible by the CLARIN Language Resource Switchboard via the portal (rather than being accessed directly).

From the LST web services portal, we have integrated the Ucto tokenizer, the Alpino dependency parser, T-Scan (text analytics), Oersetter (machine translation), fowlt (spelling correction), Frog (NLP suite for Dutch), FoLiA-stats (n-gramming), Valkuil, and Colibri Core (n-gramming). Again, these web services are accessible by the CLARIN Language Resource Switchboard via the portal.

6 Conclusion

In this deliverable, we have reported work on the integration of new web services in the workflow engine WebLicht: it now also offers easy chains to process French, Italian, Czech and Turkish texts. With the documentation of WebLicht being already in good shape, we only added a developers' checklist. The checklist guides them through the main phases of making available their web service through WebLicht.
Services from the CLARIN-DK Toolbox were also made available through the CLARIN Language Resource Switchboard, together with other web services from the CLARIN community. We have also contributed a number of web services that help converting CMDI-based metadata to bibliographic metadata such as Dublin Core, MARC 21, or MODS. Those web services are now also accessible via the Switchboard.

References

[1] Erhard W. Hinrichs, Marie Hinrichs and Thomas Zastrow. 2010. WebLicht: Web-Based LRT Services for German. In: *Proceedings of the ACL 2010 System Demonstrations*. pages 25–29.

[2] Claus Zinn *et al.* 2016. The CLARIN Language Resource Switchboard. Paper submitted to the CLARIN 2016 Annual Conference, Aix-en-Provence.

[3] S. Kaminski, T. Trippel and C. Zinn. Crosswalking from CMDI to Dublin Core and MARC 21. LREC, Portorož, Slovenia, 2016, European Language Resources Association (ELRA)

[4] T. Trippel and C. Zinn. Enhancing the Quality of Metadata by using Authority Control, 5th Workshop on Linked Data in Linguistics. Managing, Building and Using Linked Language Resources, Portorož, Slovenia, 24th May 2016. Co-located with LREC 2016, ELRA, 2016.

[5] E. Dima, E. Hinrichs, M. Hinrichs, A. Kislev, T. Trippel, T. Zastrow. 2012. Integration of WebLicht into the CLARIN Infrastructure. In *Proceedings of the joint CLARIN-D/DARIAH Workshop "Service-oriented Architectures (SOAs) for the Humanities: Solutions and Impacts" at Digital Humanities Conference 2012*. 2012, pages 17–23.

[6] TTNWW - TST Tools voor het Nederlands als Webservices in een Workflow, see https://portal.clarin.nl/node/1964 .

[7] The CLARIN-DK Tool box, see http://clarin.dk/ .

Bibliographic metadata standards:

[DC] The Dublin Core Metadata Set, see http://dublincore.org/documents/dcmi-terms/

[EAD] Encoded Archival Description, see http://www.loc.gov/ead/

[MARC] The MARC 21 Format for Bibliographic Data, see https://www.loc.gov/marc/bibliographic/

[MODS] The Metadata Object Description Schema, see http://www.loc.gov/standards/mods/

[MADS] The Metadata Authority Description Schema, see http://www.loc.gov/standards/mads/