Transatlantic Collaboration between LAPPS and CLARIN
M12 (7-12 months) Milestone Report

WP2, Tasks T2.2 and T2.3

Webservice Metadata, Communication Protocol

1. Introduction

In the first reporting period (1-6 months), we analyzed various aspects of the data converters that convert between the LAPPS Interchange Format (LIF), which is used by LAPPS services, and TCF, used by WebLicht services. We outlined the underlying structure of LIF and TCF along with JSON-LD and XML. We developed prototype software for converting documents in both directions. Options for representing tool metadata were also reviewed.
This document reports on the progress in WP2 in months 7-12 of the project. The goal in this phase was to make services in each infrastructure visible to the other infrastructure, thereby enabling invocation of all services from both sides. The main challenges in this phase were to convert the service metadata (to make services visible to the other infrastructure) and to develop a proxy service to convert between the differing communication protocols used in each infrastructure (to enable invocation of services). The following list summarized the steps which were taken to achieve the goals in this phase:

- The LIF<->TCF converters developed in first reporting period have been made available as web services.
- A web service was developed to deliver metadata of tools (detail section 2.2) on both sides on request. The retrieved metadata from the tool is then further edited using metadata editing tools (i.e. COMET).
- The repositories of both frameworks are reviewed and some service metadata of each framework are stored to the other (detail section 2.3). Services of each framework are thereby made visible to the other.
- The two infrastructures use differing communication protocols. LAPPS Grid uses the Simple Object Access Protocol (SOAP)[1] but WebLicht web services are implemented as Representational State Transfer (REST)[2] web services. A proxy (detail section 2.4) was developed, which converts a REST service request (of WebLicht) to SOAP message (of LAPPS) which is subsequently fetched by the LAPPS framework.
- The WebLicht framework had to be modified to recognize the LIF format so that the LAPPS services are selectable in the WebLicht user interface. For LAPPS side, a universal WebLicht wrapper for galaxy workflow engine is in development to further provide a consistent web-interface.

[1] https://www.w3.org/TR/soap12/
[2] https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

- Processing pipelines, including services from both frameworks, were tested (detail section 3).

The remainder of the report details various aspects of the integration of LAPPS and WebLicht (discussed above) and it is organized as follows: section two focuses on metadata structure, conversion, repository, and communication protocol. Section three presents a working example of constructing processing pipeline using services of both platforms from each framework.

2. Tool Metadata

The metadata provides all information needed in order to interact with a tool. This information makes it possible to invoke a service successfully, providing it with the data it needs for processing, and enabling interpretation of the results. WebLicht and LAPPS Grid use different methods for storing and fetching service metadata. In order to integrate the two systems, making it possible for both systems to invoke their combined set of tools, it is necessary for each system to have access to the web service metadata of the other. In this section, the metadata structure, storing, conversion, communication protocol, and the fetching process is detailed.

2.1 Metadata Structure

In order to map the service metadata from one framework to another, it is necessary to understand the underlying structure used in each format. LAPPS Grid uses the Simple Object Access Protocol (SOAP), a messaging protocol in which each web service provides its own metadata. LAPPS Grid services provide the following information:

- General information about the tool (name, description, vendor, licensing)
- Input requirements (data type, language and encoding, required previous annotations)
- Output produced (data type, language and encoding, output annotations)

The WebLicht services are stored using CMDI[3], a framework developed within CLARIN that provides a way to describe and reuse metadata components. Components are building blocks of information which can be grouped to form profiles. A Profile serves as a schema for metadata records. WebLicht services use the WebLichtWebService[4] *Profile*, which includes two types of information: *General Information* and *Orchestration Information*.

- The *General Information* contains information about creators, access rights, development status, service description, and PID (a unique ID for a web service).
- The *Orchestration Information* contains information needed to invoke the service, such as input requirements and output description.

[3] https://www.clarin.eu/content/component-metadata

[4] https://catalog.clarin.eu/ds/ComponentRegistry/#/?_k=96wfwx

2.2 Metadata Conversion

Mapping WebLicht service metadata to the LAPPS metadata format was done automatically, which was possible because all of the information required to create LAPPS service metadata can be extracted from the WebLichtWebService metadata. A web service was developed that fetches WebLicht metadata and converts it to the LAPPS metadata format.

Mapping LAPPS service metadata to WebLichtWebService metadata requires corresponding LAPPS Grid metadata as well as additional information (such as creators, the short and long description of the service, development status, etc.) relevant to identify and describe the tool in CLARIN CMDI framework. To enter this additional information for each LAPPS service, we used the CMDI Orchestration Metadata Editing Tool (COMET[5]), which is a tool for creating, editing, and validating WebLicht service metadata. The metadata is then added to the CLARIN repository for subsequent harvesting by WebLicht.

Figure 1 shows an example of metadata of Stanford Tokenizer of LAPPS (on the top) and its CMDI format (on the bottom). Similarly, Figure 2 shows an example of Stanford Tokenizer of WebLicht (on the top) and its LAPPS format (on the bottom).

Name	org.anc.lapps.stanford.Tokenizer
URL	http://vassar.lappsgrid.org/invoker/anc:stanford.tokenizer_2.0.0
Version	2.0.0
Description	Stanford Tokenizer
Vendor	http://www.anc.org
Allow	http://vocab.lappsgrid.org/ns/allow#any

Requirements

Language	en
Formats	http://vocab.lappsgrid.org/ns/media/jsonld#lif

Produces

Language	en
Formats	http://vocab.lappsgrid.org/ns/media/jsonld#lif
Annotations	http://vocab.lappsgrid.org/Token

Lapps(V): Stanford Tokenizer Stanford Tokenizer Lapps(V)

> Stanford Tokenizer Lapps(V). This is tokenizer of Lapps. It is located in Vassar Server.

PID	http://hdl.handle.net/11022/0000-0000-2518-LAPPS		
URL	http://api.lappsgrid.org/soap-proxy?id=anc:stanford.tokenizer_2.1.0-SNAPSHOT		
Email	wlsupport@sfs.uni-tuebingen.de	**Status**	development
Created	2014-07-07T11:45:58.789+02:00	**Modified**	2017-06-22T16:04:31.336Z
Input		**Output**	
type	application/json	tokens	
text			
lang	en		

[5] http://weblicht.sfs.uni-tuebingen.de/comet/

Figure 1 An example of service metadata of Stanford Tokenizer of LAPPS (on the top) and its CMDI format (on the bottom).

Stanford Tokenizer Stanford Tokenizer is a an efficient, fast, deterministic tokenizer.

Stanford Tokenizer is a an efficient, fast, deterministic tokenizer.

PID	http://hdl.handle.net/11022/0000-0000-2518-C		
Email	wlsupport@sfs.uni-tuebingen.de	**Status**	production
Created	2014-07-07T11:45:58.789+02:00	**Modified**	2014-07-07T17:11:03.775+02:00
Input		**Output**	
type	text/tcf+xml	sentences	
version	0.4	tokens	
text			
lang	en		

Name	Weblicht Stanford Tokenizer
URL	http://vassar.lappsgrid.org/invoker/anc:weblicht.stanford.tokenizer_1.0.0
Version	1.0.0
Description	LAPPS Grid wrapper around the Weblicht Stanford Tokenizer.
Vendor	http://weblicht.sfs.uni-tuebingen.de/
Allow	http://vocab.lappsgrid.org/ns/allow#any

Requirements

Encoding	UTF-8
Language	en
Formats	http://vocab.lappsgrid.org/ns/media/xml#tcf

Produces

Encoding	UTF-8
Language	en
Formats	http://vocab.lappsgrid.org/ns/media/xml#tcf
Annotations	http://vocab.lappsgrid.org/Token http://vocab.lappsgrid.org/Sentence

Figure 2 An example of Stanford Tokenizer of WebLicht (on the top) and its LAPPS format (on the bottom).

2.3 Metadata Storage and Harvesting:

Once the tool metadata has been created, it must be stored in such a way that it is accessible to the framework for which it was created.

WebLicht service metadata is harvested from CLARIN repositories. The LAPPS service metadata, which was converted to the WebLichtWebService profile format, are stored in the Tübingen CLARIN repository.

Similarly, the metadata which was converted from the WebLichtWebService profile format to the LAPPS metadata format, are stored on Brandeis University[6] and Vassar College[7] servers.

[6] http://api.lappsgrid.org/services/brandeis

[7] http://api.lappsgrid.org/services/vassar

Once the service metadata was stored in the respective repositories, it is harvested and handled within the respective frameworks in the same way as all other service metadata.

2.4 Accessing Services:

The final challenge in WP2 in this phase of the project was to actually invoke services of one framework from the other. LAPPS and WebLicht use the different communication protocols to access the services. LAPPS Grid uses SOAP, a standard messaging protocol which codifies the use of XML as an encoding scheme for request and response parameters using HTTP as a means for transport. In contrast, WebLicht web services are implemented as RESTful web services. The input to the service is sent over the web via the POST method of the HTTP protocol. The output of the webservice is the response to that POST event.

To invoke LAPPS services registered in WebLicht, it is necessary to convert WebLicht's RESTful requests to LAPPS SOAP requests. For this purpose we implemented a SOAP-PROXY that takes a REST service request as input, converts it to a SOAP message, invokes the service with the SOAP request, and returns the response from the service.

The situation is much simpler when LAPPS invokes WebLicht services as they have RESTful entry points that can be accessed via plain HTTP requests.

Figure 3 shows the mechanism of accessing services of one framework registered in the other. When WebLicht calls a LAPPS service it sends a POST request to the SOAP-PROXY which in turn makes a SOAP call to the LAPPS service. The data returned by the LAPPS SOAP service is then returned to WebLicht as the response to the POST request.

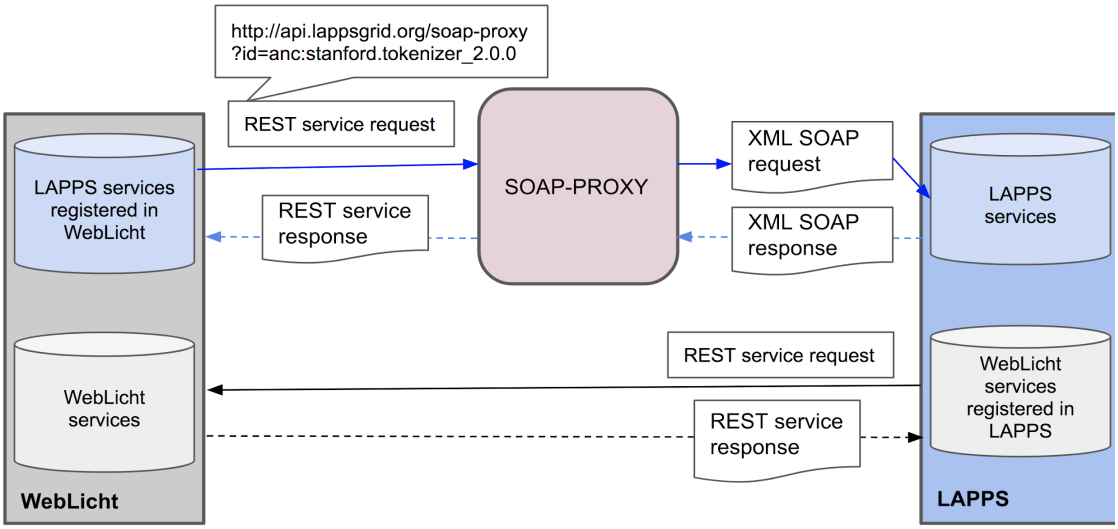

Figure 3 Accessing services of one framework registered in the other

3. State of the Integration

In the previous sections, we detail the integration of services from one framework into the other and how they can be invoked. In this section, we will show the current state of the integration efforts.

3.1 Accessing LAPPS services from WebLicht

LAPPS services are integrated into the WebLicht framework and it is possible to create and execute processing pipelines which include services from both infrastructures. Here we present a working example from the WebLicht user interface. In this example, an input text corpus is converted to the LIF format, is then tokenized and sentence-split by LAPPS services, followed by a LIF->TCF format conversion. Processing is then continued using WebLicht services.

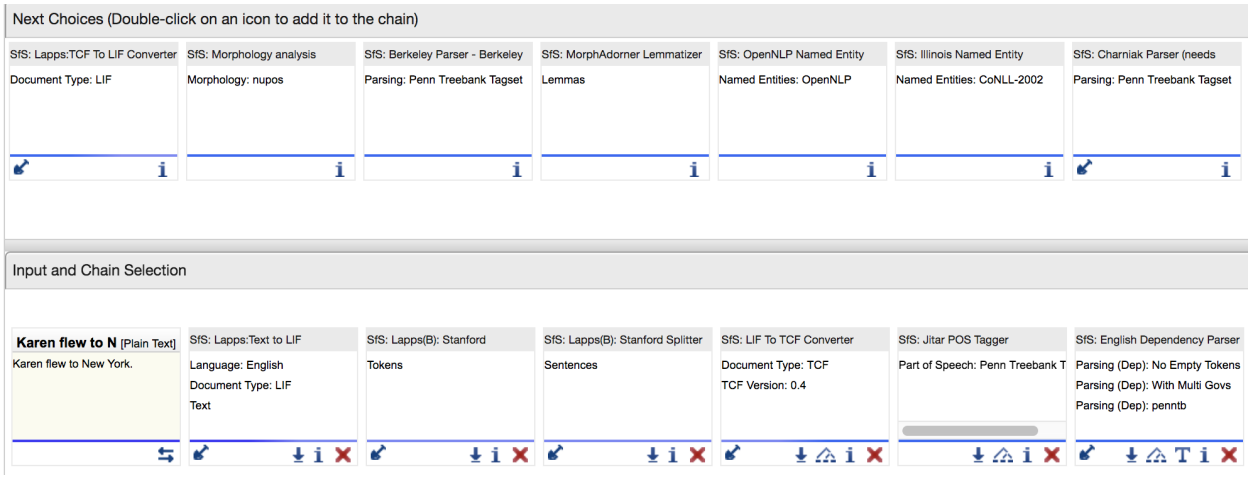

Figure 4: An example of WebLicht user interface for accessing LAPPS services from WebLicht.

As can be seen from Figure 4, a screenshot of the WebLicht user interface, the lower window 'Input and chain Selection' shows the tool chain that was selected for execution. The tool chain contains both LAPPS services and WebLicht services. After the LAPPS services (Stanford Tokenizer and Stanford Splitter) are executed, LIF → TCF converter is used to switch from LAPPS to WebLicht. The upper window ('Next Choices') shows the available WebLicht services for next selections. If TCF → LIF is chosen then the chain will switch again from WebLicht to LAPPS. In this way, a user can switch from LAPPS to WebLicht services and vice versa.

3.2 Accessing Weblicht services from LAPPS

WebLicht services have also been integrated into the LAPPS Grid framework and it is possible to create and execute processing pipelines which include services from both infrastructures on the LAPPS Grid as well. Here we present a working example from the LAPPS/Galaxy user interface. In this example, an input text document is converted to the TCF format, is then tokenized and sentence-split by WebLicht services, followed by a TCF->LIF format conversion. Processing is then continued using LAPPS Grid services.

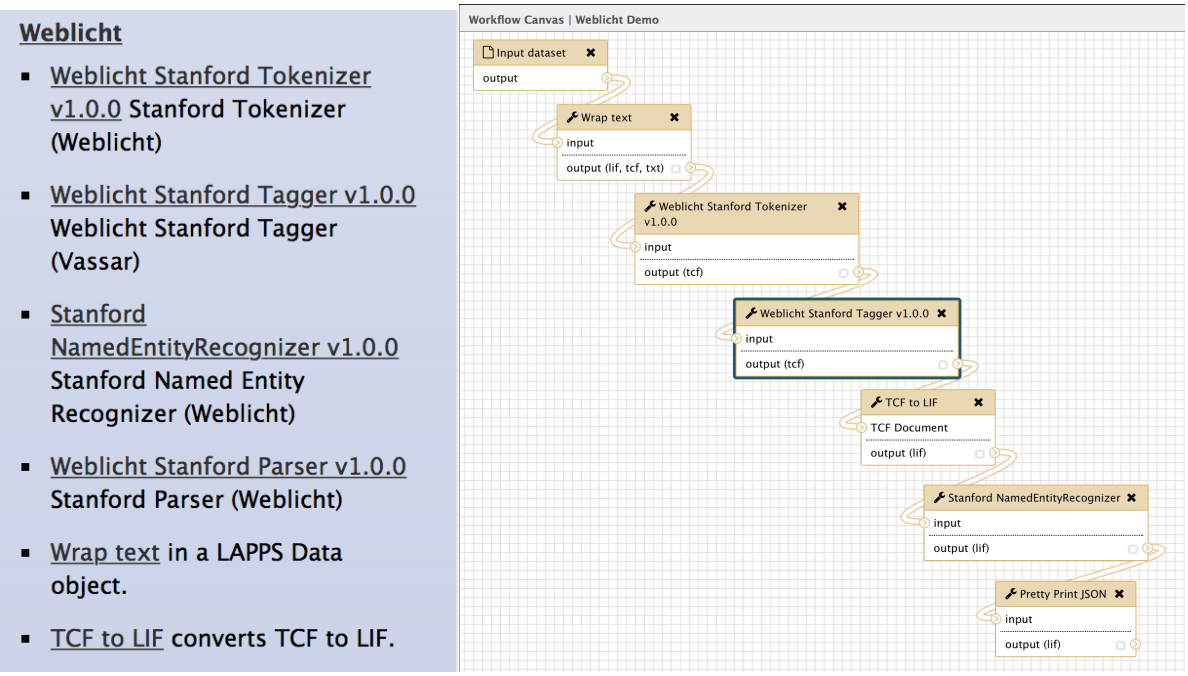

Figure 5. WebLicht services available on the LAPPS/Galaxy server and an example Galaxy workflow constructed using services from both platforms.

Figure 5 shows the Galaxy tool menu with the available WebLicht services on the left and on the right the workflow constructed using both LAPPS Grid services and WebLicht services. In this example a plain text document has been uploaded into Galaxy so the first step is to wrap the plain text as a TCF document. The TCF document is then processed by the Stanford tokenizer and part of speech tagger WebLicht services, the TCF is converted to LIF and then processed by the Stanford NamedEntityRecognizer service running on the LAPPS Grid.