LIRICS

Deliverable D.5.1.D

API for syntactic annotations

Project reference number	e-Content-22236-LIRICS
Project acronym	LIRICS
Project full title	Linguistic Infrastructure for Interoperable Resource and Systems
Project contact point	Laurent Romary, INRIA-Loria 615, rue du jardin botanique BP101. 54602 Villers lès Nancy (France) romary@loria.fr
Project web site	http://lirics.loria.fr
EC project officer	Erwin Valentini
Document title	API for morpho-syntactic annotations
Deliverable ID	D.5.1.D
Document type	Report
Dissemination level	Public
Contractual date of delivery	M18
Actual date of delivery	30/06/2006
Status & version	1.0
Work package, task & deliverable responsible	WP5 - UFSD
Author(s) & affiliation(s)	Julien Nioche, Kalina Bontcheva, Adam Funk (UFSD)
Additional contributor(s)	Niraj Aswani, Ian Roberts (UFSD)
Keywords	API SYNAF

Document evolution

version	date	version	date
0.1	06-04-06		
1.0	30-06-06		

Content

1 Introduction

This document specifies the Application Programming Interface (API) for the **Syntactic Annotation Framework** (**SYNAF**). It is currently based on the SYNAF norm given in LIRICS Deliverable 3.1 ("Evaluation of initiatives for morpho-syntactic and syntactic annotation") and ISO TC37/SC4 WD24615 ("Language Resource Management—Syntactic Annotation Framework").

This document follows the guidelines of the LIRICS deliverable D1.1 ("Guidelines and tools for producing standards, test-suites and API's").

2 Overview of the API

The LIRICS deliverable D1.1 ("Guidelines and tools for producing standards, test-suites and API's") distinguishes between two levels of API:

- **Service API** for interactions with a processor (i.e. web services, local program)
- **Linguistic Content API** for manipulating the linguistic content returned by a processor

2.1 API use cases

In this chapter we describe how the SYNAF API fits in different architectures.

2.1.1 Local access to a native SYNAF system

In this case the SYNAF system is directly accessible by the user application. Both programs are running on the same machine and are using the same implementation language. The SYNAF system runs as a library within the user application. The user application sends a representation of the document to be processed and gets in return an instance of the SYNAF **Linguistic API** objects in the implementation language of the user application. This approach is shown in Figure 1.

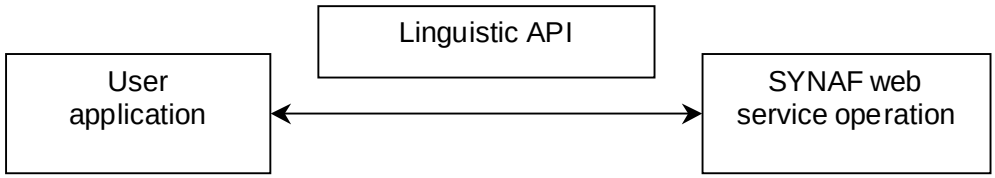

Figure 1 Direct access

2.1.2 Connection to a SYNAF Web Service

The SYNAF system runs on a remote server and is accessible through a Web Service. This approach is the one chosen in the context of the LIRICS project. The reference implementation will be remote-access-based.

A WSDL file describes the syntax and location of the annotation service. The user application sends a representation of a document to this service. The information returned can be of two types:

1. **Serialised objects:** the annotation service returns a set of serialized objects. These objects implement the **SYNAF Linguistic API**. The user application de-serialises the objects and uses them directly as in 2.1.1. This approach is not chosen in the LIRICS

project, since it requires that the user application and service use the same implementation language, which is not a portable solution.

2. **XML serialisation:** the annotation service returns a XML representation of the SYNAF document. This approach is illustrated by the Figure 2 **.**

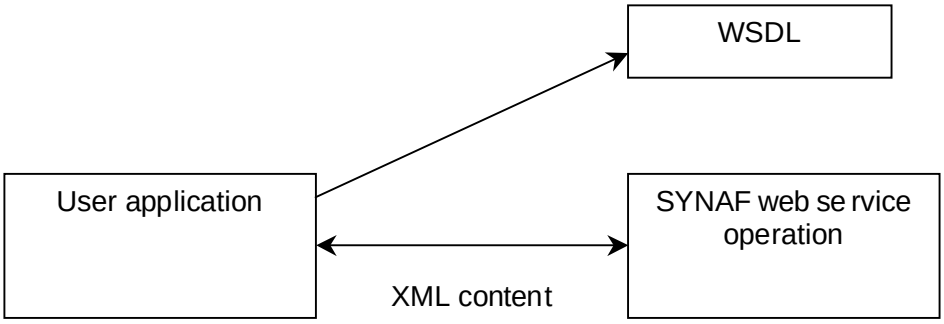

Figure 2 Remote access based on XML

The user application developer can choose whether to use this XML representation directly or to convert it into a SYNAF Linguistic API implementation. The latter requires a conversion library. After the conversion the SYNAF information can be accessed directly as in 2.1.1. This approach is illustrated by the Figure 3.

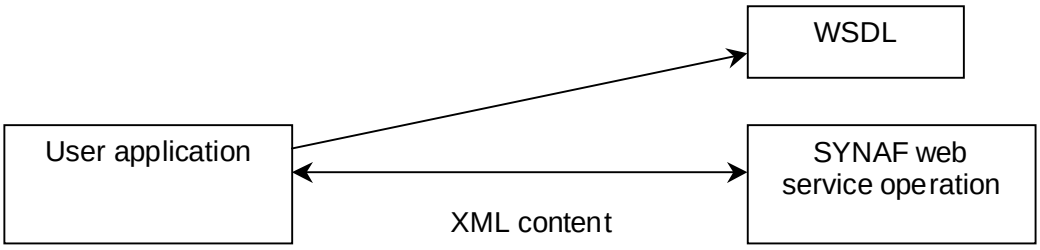

Figure 3 Remote access with conversion to a linguistic API

2.2 API strategy in LIRICS

The SYNAF linguistic API can be defined on the basis of the SYNAF model and implemented in a given language. As described in the sections above, instances of a linguistic API can be obtained:

1. directly from a SYNAF compliant system, provided that the system is loaded by the user code (see 2.1.1);

2. from a web-service returning serialized objects in the same programming language as the user code;

3. after converting the XML information returned by a web service.

In the context of the LIRICS project the SYNAF API is specified as a **service API**. Thus it will be language-independent and will be defined on the basis of web services in order to support distributed NLP resources. Dealing with XML content also leaves more choice to the client application in handling the content.

3 References:

- **Syntactic Annotation Framework** (**MAF**) ISO TC 37/SC 4 document

- LIRICS deliverable D1.1 **Guidelines and tools for producing standards, test-suites and API's**

- **Web Service Definition Language** (**WSDL**) *http://www.w3.org/TR/wsdl*

- **ISO 639-2** *http://www.id3.org/iso639-2.html*

4 SYNAF Service API

As specified in the LIRICS deliverable D1.1, the service API is described with a WSDL file. The WSDL for SYNAF Services is provided in Annex A.

The Web Services Description Language (WSDL) is an XML format for describing network services as a set of operations on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete operations are combined into services.

The content of the messages is the lexical information returned by a SYNAF server and will be in XML format. The XML format of this information is described in the SYNAF norm and reproduced in **Error! Reference source not found.**. In the WSDL file, references to the SYNAF XML elements are done through their namespace only. This way, the independence between the API description and the XML representation of SYNAF is preserved.

The SYNAF service API is based on procedures and has 5 operations:

4.1 getSupportedLanguages

This operation returns an array of character strings indicating the languages supported by the SYNAF service. The codes returned are compliant with ISO 639-2 and are 3-character long and lowercase.

The code for English, for instance, is *'eng'*, French is *'fre'*, German is *'ger'*.

4.2 getTextInputSupported

This operation returns a boolean value: *true* if and only if the SYNAF server accepts as input unannotated plain text.

4.3 getMAFInputSupported

This operation returns a boolean value: *true* if and only if the SYNAF server accepts as input text annotated with MAF according to the MAF schema.

4.4 getAnnotationFromText

This operation takes two arguments: (1) a string containing unannotated plain text of the input document and (2) a string containing the ISO 639-2 code for the language of the document.

The character strings are represented in the SOAP XML message and thus must be in the same encoding as the SOAP message. The recommended encoding is UTF-8. However, most users are lickely to connect to a SYNAF Web Service using an existing toolkit (e.g. http://ws.apache.org/axis/ for Java) which will handle the encoding automatically.

The operation returns an XML representation of an annotated SYNAF document encoded as a character string. This document will be compliant with the schema specified in the SYNAF norm.

4.5 getAnnotationFromMAF

This operation takes two arguments: (1) the MAF-annonated document, encoded as a character string and (2) a string containing the ISO 639-2 code for the language of the document.

The character strings are represented in the SOAP XML message and thus must be in the same encoding as the SOAP message. The recommended encoding is UTF-8. However, most users are lickely to connect to a SYNAF Web Service using an existing toolkit (e.g. http://ws.apache.org/axis/ for Java) which will handle the encoding automatically.

The operation returns an XML representation of an annotated SYNAF document encoded as a character string. This document will be compliant with the schema specified in the SYNAF norm.

This operation returns an error if the first input string cannot be decoded into a valid MAF XML document.

Annex A SYNAF Service WSDL description

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- WSDL definition for SYNAF annotation service -->
<wsdl:definitions targetNamespace="urn:SYNAFService"
                  xmlns:synaf="urn:SYNAFService"
                  xmlns:apachesoap="http://xml.apache.org/xml-soap"
                  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema"
            targetNamespace="urn:SYNAFService">
     <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
     <complexType name="ArrayOf_soapenc_string">
      <complexContent>
       <restriction base="soapenc:Array">
        <attribute ref="soapenc:arrayType" wsdl:arrayType="soapenc:string[]"/>
       </restriction>
      </complexContent>
     </complexType>
     <complexType name="UnsupportedLanguageException">
      <sequence/>
     </complexType>
     <complexType name="SYNAFServiceException">
      <sequence/>
     </complexType>
    </schema>
   </wsdl:types>

   <!--===== Messages =====-->

   <!-- getSupportedLanguages - takes no input and returns an array of language
        codes -->
   <wsdl:message name="getSupportedLanguagesRequest" />

   <wsdl:message name="getSupportedLanguagesResponse">
      <wsdl:part name="getSupportedLanguagesReturn"
                 type="synaf:ArrayOf_soapenc_string"/>
   </wsdl:message>

   <!-- getTextInputSupported - takes no input and returns a boolean, true if
        the service can take input from plain text, false if it cannot. -->
   <wsdl:message name="getTextInputSupportedRequest" />

   <wsdl:message name="getTextInputSupportedResponse">
      <wsdl:part name="getTextInputSupportedReturn"
                 type="xsd:boolean"/>
   </wsdl:message>

   <!-- getMAFInputSupported - takes no input and returns a boolean, true if
        the service can take input from MAF-annotated text, false if it
        cannot. -->
   <wsdl:message name="getMAFInputSupportedRequest" />

   <wsdl:message name="getMAFInputSupportedResponse">
      <wsdl:part name="getMAFInputSupportedReturn" type="xsd:boolean"/>
   </wsdl:message>

   <!-- getAnnotationFromText - takes a plain text and a language code, and
        returns the document annotated according to the SYNAF schema, as a
        string.  -->
   <wsdl:message name="getAnnotationFromTextRequest">
      <wsdl:part name="text" type="soapenc:string"/>
      <wsdl:part name="language" type="soapenc:string"/>
   </wsdl:message>

   <wsdl:message name="getAnnotationFromTextResponse">
      <wsdl:part name="getAnnotationFromTextReturn" type="soapenc:string"/>
   </wsdl:message>
```

```
<!-- getAnnotationFromMAF - takes a MAF-annotated text and a language code,
     and returns the document annotated according to the SYNAF schema, as a
     string.  -->
<wsdl:message name="getAnnotationFromMAFRequest">
   <wsdl:part name="mafDocument" type="soapenc:string"/>
   <wsdl:part name="language" type="soapenc:string"/>
</wsdl:message>

<wsdl:message name="getAnnotationFromMAFResponse">
   <wsdl:part name="getAnnotationFromMAFReturn" type="soapenc:string"/>
</wsdl:message>

<!--===== Faults =====-->

<!-- SYNAFServiceException - signals a problem when calling the service,
     typically that either an unsupported operation has been called (e.g.
     getAnnotationFromText when getTextInputSupported returns false), or
     that an internal error occurred in the service. -->
<wsdl:message name="SYNAFServiceException">
   <wsdl:part name="fault" type="synaf:SYNAFServiceException"/>
</wsdl:message>

<!-- UnsupportedLanguageException - signals that the language code passed to
     getAnnotationFrom{Text/MAF} is not supported by this service. -->
<wsdl:message name="UnsupportedLanguageException">
   <wsdl:part name="fault" type="synaf:UnsupportedLanguageException"/>
</wsdl:message>

<!--===== Service definition =====-->
<wsdl:portType name="SYNAFService">
   <wsdl:operation name="getSupportedLanguages">
      <wsdl:input name="getSupportedLanguagesRequest"
                  message="synaf:getSupportedLanguagesRequest"/>
      <wsdl:output name="getSupportedLanguagesResponse"
                   message="synaf:getSupportedLanguagesResponse"/>
   </wsdl:operation>

   <wsdl:operation name="getTextInputSupported">
      <wsdl:input name="getTextInputSupportedRequest"
                  message="synaf:getTextInputSupportedRequest"/>
      <wsdl:output name="getTextInputSupportedResponse"
                   message="synaf:getTextInputSupportedResponse"/>
   </wsdl:operation>

   <wsdl:operation name="getMAFInputSupported">
      <wsdl:input name="getMAFInputSupportedRequest"
                  message="synaf:getMAFInputSupportedRequest"/>
      <wsdl:output name="getMAFInputSupportedResponse"
                   message="synaf:getMAFInputSupportedResponse"/>
   </wsdl:operation>

   <wsdl:operation name="getAnnotationFromText"
                   parameterOrder="text language">
      <wsdl:input name="getAnnotationFromTextRequest"
                  message="synaf:getAnnotationFromTextRequest"/>
      <wsdl:output name="getAnnotationFromTextResponse"
                   message="synaf:getAnnotationFromTextResponse"/>
      <wsdl:fault name="UnsupportedLanguageException"
                  message="synaf:UnsupportedLanguageException"/>
      <wsdl:fault name="SYNAFServiceException"
                  message="synaf:SYNAFServiceException"/>
   </wsdl:operation>

   <wsdl:operation name="getAnnotationFromMAF"
                   parameterOrder="mafDocument language">
      <wsdl:input name="getAnnotationFromMAFRequest"
                  message="synaf:getAnnotationFromMAFRequest"/>
      <wsdl:output name="getAnnotationFromMAFResponse"
                   message="synaf:getAnnotationFromMAFResponse"/>
      <wsdl:fault name="UnsupportedLanguageException"
                  message="synaf:UnsupportedLanguageException"/>
      <wsdl:fault name="SYNAFServiceException"
                  message="synaf:SYNAFServiceException"/>
   </wsdl:operation>

</wsdl:portType>
```

```
<!--===== SOAP binding for the service definition =====-->
<wsdl:binding name="SYNAFServiceSoapBinding" type="synaf:SYNAFService">
   <wsdlsoap:binding style="rpc"
                     transport="http://schemas.xmlsoap.org/soap/http"/>

   <wsdl:operation name="getSupportedLanguages">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="getSupportedLanguagesRequest">
         <wsdlsoap:body use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
      </wsdl:input>
      <wsdl:output name="getSupportedLanguagesResponse">
         <wsdlsoap:body use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
      </wsdl:output>
   </wsdl:operation>

   <wsdl:operation name="getTextInputSupported">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="getTextInputSupportedRequest">
         <wsdlsoap:body use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
      </wsdl:input>
      <wsdl:output name="getTextInputSupportedResponse">
         <wsdlsoap:body use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
      </wsdl:output>
   </wsdl:operation>

   <wsdl:operation name="getMAFInputSupported">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="getMAFInputSupportedRequest">
         <wsdlsoap:body use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
      </wsdl:input>
      <wsdl:output name="getMAFInputSupportedResponse">
         <wsdlsoap:body use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
      </wsdl:output>
   </wsdl:operation>

   <wsdl:operation name="getAnnotationFromText">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="getAnnotationFromTextRequest">
         <wsdlsoap:body use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
      </wsdl:input>
      <wsdl:output name="getAnnotationFromTextResponse">
         <wsdlsoap:body use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
      </wsdl:output>
      <wsdl:fault name="UnsupportedLanguageException">
         <wsdlsoap:fault name="UnsupportedLanguageException" use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
      </wsdl:fault>
      <wsdl:fault name="SYNAFServiceException">
         <wsdlsoap:fault name="SYNAFServiceException" use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
      </wsdl:fault>
   </wsdl:operation>

   <wsdl:operation name="getAnnotationFromMAF">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="getAnnotationFromMAFRequest">
         <wsdlsoap:body use="encoded"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="urn:SYNAFService"/>
```

```
            </wsdl:input>
            <wsdl:output name="getAnnotationFromMAFResponse">
               <wsdlsoap:body use="encoded"
                 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="urn:SYNAFService"/>
            </wsdl:output>
            <wsdl:fault name="UnsupportedLanguageException">
               <wsdlsoap:fault name="UnsupportedLanguageException" use="encoded"
                 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="urn:SYNAFService"/>
            </wsdl:fault>
            <wsdl:fault name="SYNAFServiceException">
               <wsdlsoap:fault name="SYNAFServiceException" use="encoded"
                 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="urn:SYNAFService"/>
            </wsdl:fault>
         </wsdl:operation>
      </wsdl:binding>
   </wsdl:definitions>
```