Language Grid
NICT Language Grid Project

How to Use Language Services

2009/2/19

NICT Language Grid Project

http://langrid.nict.go.jp

Contents

1. Introduction

The language gird is a multilingual service infrastructure developed by the National Institute of Information and Communications Technology's Language Grid Project, which, by coordinating language resources such as machine translation and dictionaries available on the internet, lays the foundation for new language services. Coordinating multiple machine translations, for instance, allows users to translate between languages that have never before been machine-translatable, and aligning specialized bilingual dictionaries with machine translation establishes machine translations with even better specialized capabilities.

This manual outlines usage methods for this language grid.

1.1. Who this Manual is Intended for

The language grid includes the following three types of stakeholders.

- Language service users
  Users who invoke/call and utilize various language services registered on the language grid.
- Language resource providers
  Users who provide their own language resources to the language grid
- Computational resource providers
  Users who provide either or both language grid core nodes / language grid service nodes, which form the language grid.

This manual explains usage methods for the language grid and is directed especially towards language service users. These language service users include developers of collaboration tools that use language grid language services, operators who support collaboration with end users managing those collaboration tools, as well as end users who use collaboration tools to collaborate with people abroad. Of these, this manual focuses particularly on collaboration tool developers and explains related language grid usage methods under the assumption that the reader has knowledge of programming and web service technologies. For more information on programming and web service technologies, please refer to the separate text and/or information available on the web (Services Computing School: http://langrid.org/services/jp/index.html, etc.).

1.2. Language Grid Provisioning Features

The language grid provides the following functions to language service users.

- Acquisition of language service information
  Searches language services stored on the language grid, based on service profile information, and provides profile information and WSDL information as results.
- Execution of language services

Based on given input, executes translation services, bilingual dictionary services, bilingual dictionary headline extraction services, conceptual dictionary services, example translation services, morphological analysis services, paraphrase services, similarity calculation services, adjacency-response services, dependency parsing services, back translation services, and multi-hop translation services and provides results.

– Dynamic binding

  This function allows specification of invoked language services in the execution of compound services. This enables specification of services used in back translation compound services on the caller side. Hierarchical specification of bindings in invoking nested compound services is possible in dynamic binding (hierarchical binding).

– Return functions for invocation/call information

  A function that enables the return of copyright notices for invoked services and call trees upon execution of compound services (items that hierarchically express which service was actually called from which service).

1.3. Detailed Interface Information

For detailed interface information, please refer to the following JavaDoc-supported classes.

Japanese: http://langrid.nict.go.jp/developer/ja/apidocs/

English: http://langrid.nict.go.jp/developer/en/apidocs/

2. Language Service Information Acquisition API

The language service provides a variety of API that manage the services. This chapter explains API that perform service searches and service information acquisition.

2.1. WSDL and Call Target URL

Setting the core node URL to ${CORENODEURL} allows you to use the following URLs to perform WSDL acquisition for or call the services explained in this chapter.

WSDL acquisition: ${CORENODEURL}/services/ServiceManagement?wsdl
Service call/invocation: ${CORENODEURL}/services/ServiceManagement

Below is the URL for the core node provided by the NICT Language Grid Project. To acquire or execute WSDL, a language grid user ID and password are required.

http://langrid.nict.go.jp/langrid-1.2

2.2. Management Service Interface (for Information Acquisition only)

API for language service information acquisition are shown below. For more information, please refer to the jp.go.nict.langrid.foundation.servicemanagement.ServiceManagement class JavaDoc.

Name	Method Name	Explanation
Service search	searchServices	Searches services registered on the language grid.
Service profile information acquisition	getServiceProfile	Acquires profile information for services registered on the language grid.
WSDL acquisition	getServiceWsdl	Acquires WSDL for calling services registered on the language grid.
Acquisition of external invocation information	getExternalInvocations	Acquires external calling information used in hierarchical binding. External calling refers to calling other language services from compound services. External calling information contains the call's classification name (invocationName), service type, and other information.

WSDL that can be acquired by the above WSDL acquisition API may also be acquired by accessing the following URL.

http://[address]/langrid-1.2/wsdl/[serviceId]

[address] is the address of the server where the core node is running, while[serviceId] refers to the WSDL the user would like to acquire.

3. Language Service Execution API

This chapter details API related to the execution of language services.

3.1. Common Functions

3.1.1. Dynamic Binding Specification Functions

These functions can dynamically specify services called from compound services while compound services are in use. Specifying services called from compound services is called binding. When the call target service is a compound service, it is possible to specify the binding within that service (hierarchical binding). Explanations of hierarchical binding specifications and designation

methods are shown below.

Hierarchical Binding

This allows users to make hierarchical specifications of service call bindings in compound services. When compound services are called hierarchically, it is possible to make detailed specifications about what kinds of binding are performed in which compound services. In addition, information that classifies binding uses external invocation names (in BPEL, partner links). This information can be acquired with language grid external invocation information acquisition API, and can also be viewed by service managers. Information used in binds is as follows.

```
[]
Or
[BINDINGNODE,...]

BINDINGNODE - bind information
```

Bind information is made up of the following information.

```
{
  "invocationName" : IVN
  , "serviceId" : TID
  , "children" : [] or は [BINDINGNODE,...]
}

IVN - "invocationName": Call name
TID - ID/Call target URL for the invoked service
```

Hierarchical binding describes information in JSON encoding. The following is used to carry out the following procedure:

1. Call the BackTranslation compound service;
2. Call TwoHopTranslationEn to ForwardTranslationPL;
3. Designated NICTJServer to FirstTranslationPL and NICTCLWT to SecondTranslationPL;
4. Call TwoHopTranslationEn to BackTranslation's BackwardTranslationPL;
5. Designate NICTCLWT to FirstTranslationPL and NICTJServer to SecondTranslationPL

(In this case, designate “ja” for the back translation interface sourceLang parameter and “fr” for the intermediateLang parameter. However, this code may be disabled because of service registration status and/or usage restriction(s).)

```
[{
   "invocationName":"ForwardTranslationPL"
   ,"serviceId":"TwoHopTranslationEn"
   ,"children":[{
      "invocationName":"FirstTranslationPL"
      ,"serviceId":"NICTJServer"
      ,"children":[]
   },{
      "invocationName":"SecondTranslationPL"
      ,"serviceId":"NICTCLWT"
      ,"children":[]
   }]
},{
   "invocationName":"BackwardTranslationPL"
   ,"serviceId":"TwoHopTranslationEn"
   ,"children":[{
      "invocationName":"FirstTranslationPL"
      ,"serviceId":"NICTCLWT"
      ,"children":[]
   },{
      "invocationName":"SecondTranslationPL"
      ,"serviceId":"NICTJServer"
      ,"children":[]
   }]
}]
```

※When making actual specifications, the above information must be HTML-escaped (" -> ")

SOAP Interface

One may designate bindings by including various pieces of information in the SOAP header. SOAP header namespace and an actual specification example are shown below.

SOAP Header

Inserting the following value into the requested SOAP header allows you to specify binding.

Binding	**namespace**	**Tag**
Hierarchical binding	http://langrid.nict.go.jp/process/binding/tree	binding

Specification Example

The following shows SOAP messages resulting from actually specifying bindings and performing translation. In the examples below, the following bindings are performed on

TwoHopTranslationEn.

- Hierarchical Binding

  TwoHopTranslationJa to External invocation FirstTranslationPL

  Then, in TwoHopTranslationJa

  NICTJServer to external invocation FirstTranslationPL

  NICTCLWT to external invocation SecondTranslationPL

  TwoHopTranslationJa to external invocation SecondTranslationPL

  Then, in TwoHopTranslationJa

  NICTJServer to external invocation FirstTranslationPL

  NICTCLWT to external invocation SecondTranslationPL

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns3:binding soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next" soapenv:mustUnderstand="0"
      xsi:type="soapenc:string" xmlns:ns3="http://langrid.nict.go.jp/process/binding/tree"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
      >[{
        &quot;children&quot;:[{
          &quot;children&quot;:[]
          ,&quot;invocationName&quot;:&quot;FirstTranslationPL&quot;
          ,&quot;serviceId&quot;:&quot;NICTJServer&quot;
        },{
          &quot;children&quot;:[]
          ,&quot;invocationName&quot;:&quot;SecondTranslationPL&quot;
          ,&quot;serviceId&quot;:&quot;NICTCLWT&quot;
        }]
        ,&quot;invocationName&quot;:&quot;FirstTranslationPL&quot;
        ,&quot;serviceId&quot;:&quot;TwoHopTranslationJa&quot;
      },{
        &quot;children&quot;:[{
          &quot;children&quot;:[]
          ,&quot;invocationName&quot;:&quot;FirstTranslationPL&quot;
          ,&quot;serviceId&quot;:&quot;NICTJServer&quot;
        },{
          &quot;children&quot;:[]
          ,&quot;invocationName&quot;:&quot;SecondTranslationPL&quot;
          ,&quot;serviceId&quot;:&quot;NICTCLWT&quot;
        }]
        ,&quot;invocationName&quot;:&quot;SecondTranslationPL&quot;
        ,&quot;serviceId&quot;:&quot;TwoHopTranslationJa&quot;
      }]</ns3:binding>
  </soapenv:Header>
  <soapenv:Body>
    <ns3:backTranslate soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns3="http://backtranslation.translation.ws_1_2.wrapper.langrid.nict.go.jp">
      <sourceLang xsi:type="xsd:string">en</sourceLang>
      <intermediateLang xsi:type="xsd:string">zh</intermediateLang>
```

```
        <source xsi:type="xsd:string">how are you</source>
      </ns3:backTranslate>
  </soapenv:Body>
</soapenv:Envelope>
```

However, this code may be disabled because of service registration status and/or usage restriction(s).

Java Interface

The following method is available for specifying bindings in the jp.go.nict.langrid.client.ws_1_2.ServiceClient.

```
/**
 * Get hierarchical binding information.
 * @return hierarchical binding information
 */
Collection<BindingNode> getTreeBindings();
```

Adding call name, service ID (or URL), or hierarchically applied binding (children) to the BindingNode returned by getTreeBindings allows binding specification.

A code example for carrying out the aforementioned binding appears below. Variable client indicates the language grid client. Code details are explained in chapter 4.

```
client.getTreeBindings().add(
  new BindingNode("FirstTranslationPL", "TwoHopTranslationJa")
    .addChild(new BindingNode("FirstTranslationPL", "NICTJServer"))
    .addChild(new BindingNode("SecondTranslationPL", "NICTCLWT"))
).add(
  new BindingNode("SecondTranslationPL", "TwoHopTranslationJa")
    .addChild(new BindingNode("FirstTranslationPL", "NICTJServer"))
    .addChild(new BindingNode("SecondTranslationPL", "NICTCLWT"))
);
```

3.1.2. Service Information Return Functions

When a language service is executed, this function returns the name, copyright information, and license information of the called service. Information is returned for all services called.

HTTP Interface

Service information is returned as an HTTP header. Headers and corresponding explanations are shown below.

Header	Explanation

X-LanguageGrid-ServiceName	Service name
X-LanguageGrid-ServiceCopyright	Service copyright information
X-LanguageGrid-ServiceLicense	Service license information

A returned HTTP header appears below.

```
HTTP/1.0 200 OK
Date: Thu, 17 Jul 2008 09:16:00 GMT
Content-Type: text/xml;charset=utf-8
Content-Length: 2793
X-LanguageGrid-ServiceName: Two-Hop Translation Service through English
X-LanguageGrid-ServiceCopyright: Copyright 2008 NICT Language Grid Project.
X-LanguageGrid-ServiceLicense: http://langrid.nict.go.jp/
```

Java Interface

The following methods are available for acquiring information in the jp.go.nict.langrid.client.ws_1_2.ServiceClient.

```
/**
 * The name of the most recently called service is returned.
 * @return service name
 */
String getLastName();

/**
 * The copyright information of the most recently called service is returned.
 * @return copyright information
 */
String getLastCopyrightInfo();

/**
 * The license information of the most recently called service is returned.
 * @return license information
 */
String getLastLicenseInfo();
```

3.1.3. Call Tree Return Functions

When a compound service is executed, this function returns information, in the form of a call tree, for all services executed concomitantly with the compound service. Service information

has the following construction.

```
CallNode{
  String serviceId; // Service ID
  String serviceName;  // Service name
  String serviceCopyright;  // Copyright information
  String serviceLicense;  // License information
  int responseTimeMillis;  // Response time (milliseconds)
  String faultCode;  // Fault code at failure (SOAPFault.faultCode)
  String faultString;  // Fault string at failure (SOAPFault.faultString)
  CallNode[] children;  // Call tree for when another service is
                        //  being called from the present service

}
```

SOAP Interface

Call trees are returned as SOAP headers with a namespace of http://langrid.nict.go.jp/process/calltree and a “calltree” tag. Values are the aforementioned CallNode arrays, encoded in JSON format. HTML-escaping (" -> &quot, etc.) is also performed in SOAP headers.

The following illustrates returned SOAP data when TwoHopTranslationJa (compound service) is called twice from TwoHopTranslationEn (compound service) and NICTJServer (atomic service) is called from TwoHopTranslationJa service.

```
<?xml version="1.0" encoding="UTF-8"?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <soapenv:Header>
        <ns1:calltree soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
          soapenv:mustUnderstand="0" xsi:type="soapenc:string"
          xmlns:ns1="http://langrid.nict.go.jp/process/calltree"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
        >[{
          &amp;quot;children&amp;quot;:[{
            &amp;quot;children&amp;quot;:[]
            ,&amp;quot;faultCode&amp;quot;:&amp;quot;&amp;quot;
            ,&amp;quot;faultString&amp;quot;:&amp;quot;&amp;quot;
            ,&amp;quot;responseTimeMillis&amp;quot;:261
            ,&amp;quot;serviceCopyright&amp;quot;:&amp;quot;&amp;quot;
            ,&amp;quot;serviceId&amp;quot;:&amp;quot;NICTJServer&amp;quot;
            ,&amp;quot;serviceLicense&amp;quot;:&amp;quot;&amp;quot;
            ,&amp;quot;serviceName&amp;quot;:&amp;quot;J-Server&amp;quot;
          },{
            &amp;quot;children&amp;quot;:[]
```

```
            ,&quot;faultCode&quot;:&quot;&quot;
            ,&quot;faultString&quot;:&quot;&quot;
            ,&quot;responseTimeMillis&quot;:222
            ,&quot;serviceCopyright&quot;:&quot;Cross Language Inc.&quot;
            ,&quot;serviceId&quot;:&quot;NICTCLWT&quot;
            ,&quot;serviceLicense&quot;:&quot;&quot;
            ,&quot;serviceName&quot;:&quot;WEB-Transer&quot;
          }]
          ,&quot;faultCode&quot;:&quot;&quot;
          ,&quot;faultString&quot;:&quot;&quot;
          ,&quot;responseTimeMillis&quot;:814
          ,&quot;serviceCopyright&quot;:&quot;&quot;
          ,&quot;serviceId&quot;:&quot;TwoHopTranslationJa&quot;
          ,&quot;serviceLicense&quot;:&quot;&quot;
          ,&quot;serviceName&quot;:&quot;Two-Hop Translation Service through Japanese&quot;
        },{
          &quot;children&quot;:[{
            &quot;children&quot;:[]
            ,&quot;faultCode&quot;:&quot;&quot;
            ,&quot;faultString&quot;:&quot;&quot;
            ,&quot;responseTimeMillis&quot;:280
            ,&quot;serviceCopyright&quot;:&quot;&quot;
            ,&quot;serviceId&quot;:&quot;NICTJServer&quot;
            ,&quot;serviceLicense&quot;:&quot;&quot;
            ,&quot;serviceName&quot;:&quot;J-Server&quot;
          },{
            &quot;children&quot;:[]
            ,&quot;faultCode&quot;:&quot;&quot;
            ,&quot;faultString&quot;:&quot;&quot;
            ,&quot;responseTimeMillis&quot;:323
            ,&quot;serviceCopyright&quot;:&quot;Cross Language Inc.&quot;
            ,&quot;serviceId&quot;:&quot;NICTCLWT&quot;
            ,&quot;serviceLicense&quot;:&quot;&quot;
            ,&quot;serviceName&quot;:&quot;WEB-Transer&quot;
          }]
          ,&quot;faultCode&quot;:&quot;&quot;
          ,&quot;faultString&quot;:&quot;&quot;
          ,&quot;responseTimeMillis&quot;:731
          ,&quot;serviceCopyright&quot;:&quot;&quot;
          ,&quot;serviceId&quot;:&quot;TwoHopTranslationJa&quot;
          ,&quot;serviceLicense&quot;:&quot;&quot;
          ,&quot;serviceName&quot;:&quot;Two-Hop Translation Service through Japanese&quot;
        }]</ns1:calltree>
    </soapenv:Header>
    <soapenv:Body>
       <ns2:translateResponse
         soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         xmlns:ns2="http://translation.ws_1_2.wrapper.langrid.nict.go.jp">
          <translateReturn xsi:type="xsd:string"
            >&#50504;&#45397;&#54616;&#49901;&#45768;&#44620;?</translateReturn>
       </ns2:translateResponse>
    </soapenv:Body>
  </soapenv:Envelope>
```

When an error occurs in the execution of a service, information indicating a failure is stored in the failed invocation's faultCode and faultString. As long as no other error occurs in subsequent processing, that information is preserved in the superior CallNode.

The following shows a call tree at the time of a failure. HTML-escaping is omitted.

```
[{
 "children": [{
     "responseTimeMillis": 2047,
     "serviceCopyright": "Copyright(C) 2007 NICT Language Grid Project",
     "serviceId": "NICTJServer",
     "serviceLicense": "http://langrid.nict.go.jp/",
     "serviceName": "JServer translation service provided by Kodensha Co., Ltd."
 }, {
     "faultCode": "Server.userException",
     "faultString": "jp.go.nict.langrid.service_1_2.NoValidEndpointsException: no valid endpoints for service
¥"NICTCLWT¥".",
     "responseTimeMillis": 31,
     "serviceCopyright": "",
     "serviceId": "NICTCLWT",
     "serviceLicense": "",
     "serviceName": "Cross Language Web Transer"
 }],
 "faultCode": "NoValidEndpointsException",
 "faultString": "jp.go.nict.langrid.service_1_2.NoValidEndpointsException: no valid endpoints for service
¥"NICTCLWT¥".",
 "responseTimeMillis": 2344,
 "serviceCopyright": "",
 "serviceId": "TwoHopTranslationJa",
 "serviceLicense": "",
 "serviceName": "Two-Hop Translation Service through Japanese"
}]
```

Java Interface

The Java client library contains the jp.go.nict.langrid.client.ws_1_2.ServiceClient interface, in which a method for acquiring call trees is declared. The ServiceClient is implemented with all language service clients.

```
/**
 * The call tree of the most recently called service is returned.
 * @return call tree
 */
Collection<CallNode> getLastCallTree();
```

Call trees can be acquired regardless of the rights and wrongs of service execution, but should an error occur prior to service invocation, empty information is returned.

3.2. Language Service Interface

The following contains a list of Java interfaces that define language service interfaces, along with their accompanying explanations. For more detailed interface information, please refer to the javadoc in section 1.3.

Language Service Type (type name)
Machine translation (TRANSLATION)
Java Interface
jp.go.nict.langrid.service_1_2.translation.TranslationService
Explanation
An interface that performs the machine translation service. Regulates a simple interface that collects source language, target language, and source document in arguments.

Language Service Type (type name)
Translation service interface combined with a Temporal dictionary (TRANSLATIONWITHTEMPORALDICTIONARY)
Java Interface
jp.go.nict.langrid.service_1_2.translation.TranslationWithTemporalDictionaryService
Explanation
An interface for a service that coordinates with a temporal dictionary and performs machine translation. The temporal dictionary is a low-data bilingual dictionary that is not registered on the language grid.

Language Service Type (type name)
Back translation service interface (BACKTRANSLATION)
Java Interface
jp.go.nict.langrid.service_1_2.backtranslation.BackTranslationService
Explanation
An interface for a service that performs back translation.

Language Service Type (type name)
Multi-hop translation service interface (MULTIHOPTRANSLATION)
Java Interface
jp.go.nict.langrid.service_1_2.multihoptranslation.MultihopTranslationService
Explanation
An interface for the service that performs multi-hop translation. This service translates text to the

target language via one or more languages.

Language Service Type (type name)
Bilingual dictionary service interface (BILINGUALDICTIONARY)
Java Interface
jp.go.nict.langrid.service_1_2.bilingualdictionary.BilingualDictionaryService
Explanation
An interface that displays a simple bilingual dictionary made up of only direction words and antonyms.

Language Service Type (type name)
A bilingual dictionary service interface with a longest match search function based on morphemes (BILINGUALDICTIONARYWITHLONGESTMATCHSEARCH)
Java Interface
jp.go.nict.langrid.service_1_2.bilingualdictionary. BilingualDictionaryWithLongestMatchSearchService
Explanation
In addition to all interfaces supported by the bilingual dictionary function, this language service also offers an interface for longest match searching based on morphemes.

Language Service Type (type name)
Pictogram dictionary service interface (PICTOGRAMDICTIONARY)
Java Interface
jp.go.nict.langrid.service_1_2.pictogramdictionary.PictogramDictionaryService
Explanation
A pictogram dictionary interface that acquires pictograms associated with direction words.

Language Service Type (type name)
Conceptual dictionary service interface (CONCEPTDICTIONARY)
Java Interface
jp.go.nict.langrid.service_1_2.conceptdictionary.ConceptDictionaryService
Explanation
A conceptual dictionary interface for acquiring concepts expressed by synonym clusters (Sysnet) that allows for the acquisition of concepts by searching words included in synonym clusters and pursuing

connections from a given concept.

Language Service Type (type name)
Example translation service interface (PARALLELTEXT)
Java Interface
jp.go.nict.langrid.service_1_2.paralleltext.ParallelTextService
Explanation
An example translation interface for acquiring given example translations.

Language Service Type (type name)
Adjacency-response support service interface (ADJACENCYPAIR)
Java Interface
jp.go.nict.langrid.service_1_2.adjacencypair.AdjacencyPairService
Explanation
An interface that acquires lists of responses for specified utterances.

Language Service Type (type name)
Morphological analysis service interface (MORPHOLOGICALANALYSIS)
Java Interface
jp.go.nict.langrid.service_1_2.morphologicalanalysis.MorphologicalAnalysisService
Explanation
An interface that divides input strings into morphological units.

Language Service Type (type name)
Paraphrase service interface (PARAPHRASE)
Java Interface
jp.go.nict.langrid.service_1_2.paraphrase.ParaphraseService
Explanation
An interface that rephrases input strings into other expressions.

Language Service Type (type name)
Similarity calculation service interface (SIMILARITYCALCULATION)
Java Interface
jp.go.nict.langrid.service_1_2.similaritycalculation.SimilarityCalculationService
Explanation

An interface that measures the similarity of two input-specified sentences.

Language Service Type (type name)
Dependency parsing service interface (DEPENDENCYPARSER)
Java Interface
jp.go.nict.langrid.service_1_2.dependencyparser.DependencyParserService
Explanation
An interface that separates input strings into various clusters (clauses, phrases, etc.) and analyzes the dependency between resulting clusters.

4. Java Code Examples

This chapter gives explanations of sample codes actually used in the language grid. Code in this chapter uses Java client libraries offered by the Language Grid Project.

4.1. Java Client Library Download / Setup

Formats for Java client libraries and other required libraries can be downloaded by visiting the **Wiki for language grid developers** (http://langrid.nict.go.jp/langrid-developers-wiki/), selecting various packages, and accessing client libraries (langrid-client-“date and time”.zip).

Then, decompress the archive and place all of the contained libraries in a location in the Java classpath, add the location containing the expanded .zip file to the system classpath, or specify classpath at Java execution or designate classpath via the cp option.

4.2. Using the Clients

In order to invoke language services, we create the client objects for each language service interface using the below method from the jp.go.nict.langrid.client.ws_1_2.ClientFactory class.

public static ${SERVICE_TYPE}Client create${SERVICE_TYPE}Client(URL serviceUrl);

In serviceUrl, we set the access URL of the language service being invoked (ex: http://langrid.nict.go.jp/langrid-1.2/invoker/${SERVICE_ID} **replace ${SERVICE_ID} with the ID of the invoked service). An array giving the type name of each language service goes in {SERVICE_TYPE}. For example, it would be "createTranslationClient" for a translation service client, and "createMorphologicalAnalysisClient" in the case of a client for morphological analysis.

When using Basic authentication parameters, please call the setDefaultUserId and setDefaultPassword methods of ClientFactory. All clients created through ClientFactory from then on will have user ID and password automatically set. By calling each client object's setUserId and setPassword you can change the settings individually. Otherwise, ClientFactory provides automatic network proxy recognition capability. This corresponds with the proxy information set in the Java VM and WPAD set to use DNS. You can get a list of interfaces of clients to create at the JavaDoc http://langrid.nict.go.jp/developer/ja/apidocs/jp/go/nict/langrid/client/ws_1_2/package-summary.html. All interfaces of language and administrative services are listed.

4.3. Calling Atomic Services

Sample coding that calls NICTJ Server - registered on the language grid - is shown below as an atomic service calling example.

```
package jp.go.nict.langrid.sample;

import static jp.go.nict.langrid.language.ISO639_1LanguageTags.en;
import static jp.go.nict.langrid.language.ISO639_1LanguageTags.ja;
import java.net.URL;
import jp.go.nict.langrid.client.ws_1_2.ClientFactory;
import jp.go.nict.langrid.client.ws_1_2.TranslationClient;

public class SimpleTranslation {
  public static void main(String[] args) throws Exception{
    TranslationClient c = ClientFactory.createTranslationClient(new URL(  // 1
      "http://langrid.nict.go.jp/langrid-1.2/invoker/NICTJServer"         // 1
      ));                                                                 // 1
    c.setUserId("Your ID");              // 2
    c.setPassword("Your Password");      // 2

    System.out.println("result: " + c.translate(en, ja, "How are you?")); // 3
    System.out.println("serviceName: " + c.getLastName());                // 4
    System.out.println("copyrightInfo: " + c.getLastCopyrightInfo());     // 5
    System.out.println("licenseInfo: " + c.getLastLicenseInfo());         // 6
  }
}
```

The following processes are performed in the code.

1. Creates the client which invokes the machine translation service.
2. Sets up authentication information (language grid user ID / passwords).
3. Calls the service and displays results.
4. Displays the name of the service returned from the language grid.

5. Displays the copyright information of the service returned from the language grid.
6. Displays the license information of the service returned from the language grid.

Items 4-6 are items of information that are related to the called service, stored on the language grid, and returned each time a service is called.

4.4. Calling Compound Services

Shown below is an example of a compound service invocation in which the TwoHopTranslation service is used to perform 2-hop translation (English-Japanese-Korean). Because compound services are constructed with an abstract workflow, the point that uses the dynamic binding for specifying the specific language resource at invocation differs from atomic services.

```
package jp.go.nict.langrid.sample;

import static jp.go.nict.langrid.language.ISO639_1LanguageTags.en;
import static jp.go.nict.langrid.language.ISO639_1LanguageTags.ja;
import static jp.go.nict.langrid.language.ISO639_1LanguageTags.ko;

import java.net.URL;

import jp.go.nict.langrid.client.ws_1_2.ClientFactory;
import jp.go.nict.langrid.client.ws_1_2.MultihopTranslationClient;
import jp.go.nict.langrid.commons.cs.binding.BindingNode;
import jp.go.nict.langrid.commons.cs.calltree.CallTreeUtil;
import jp.go.nict.langrid.language.Language;

public class BindingTwoHopTranslations {
  public static void main(String[] args) throws Exception{
    MultihopTranslationClient c = ClientFactory.createMultihopTranslationClient(new URL(
      "http://langrid.nict.go.jp/langrid-1.2/invoker/TwoHopTranslation"
      ));
    c.setUserId("Your ID");
    c.setPassword("Your Password");
    c.getTreeBindings().add(new BindingNode("FirstTranslationPL", "NICTJServer")); // 1
    c.getTreeBindings().add(new BindingNode("SecondTranslationPL", "NICTCLWT"));   // 1
    try{
      System.out.println(                                                  // 2
        "result: "                                                         // 2
        + c.multihopTranslate(en, new Language[]{ja}, ko, "how are you")); // 2
      System.out.println("serviceName: " + c.getLastName());
      System.out.println("copyrightInfo: " + c.getLastCopyrightInfo());
      System.out.println("licenseInfo: " + c.getLastLicenseInfo());
    } finally{
      System.out.println(                                                  // 3
        "calltree: " + CallTreeUtil.encodeTree(c.getLastCallTree(), 2)   // 3
        );                                                                 // 3
    }
  }
```

```
}
```

The following processes are performed in the code.

1. Binds the NICTJServer translation to FirstTranslationPL in the TwoHopTranslation and binds the NICTCLWT translation service to SecondTranslationPL.
2. Calls TwoHopTranslation and displays results. The TwoHopTranslation interface is the multi-hop translation interface, specifying the translation source language, intermediate language arrangement, translation target language, and the text to be translated.
3. Displays call tree.

In this sample code, Korean is displayed as the execution result, but this may not be displayed correctly depending on your development environment. In such cases, write the results to a file and display them in software with multilingual display support (web browsers, etc.) to check them.

5. Translation Combined with Bilingual Dictionary

5.1. What is Translation Combined with Bilingual Dictionary?

The dictionary coordinated translation service refers to a compound service that, by linking bilingual dictionaries and machine translations and using translations registered in the dictionary, produces translations of higher quality than those produced by machine translation only. This service comes in two types. The first type, called the translation combined with bilingual dictionary service, links bilingual dictionaries that offer normal search functions, such as complete match, partial match, front match, and back match. The second type links bilingual dictionaries that provide longest match search functions; this service is called translation combined with bilingual dictionary with longest match search. This second type uses the bilingual dictionary's longest match search, which speeds up the normal translation combined with bilingual dictionary service.

5.2. Translation Combined with Bilingual Dictionary Specifications

The translation combined with bilingual dictionary service and translation combined with bilingual dictionary with longest match search services employ a translation interface (TranslationWithTemporalDictionary), compatible with bilingual (Temporal) dictionary input, in order to also link Temporal dictionaries that have not been adapted for Web services. In addition to the information required for machine translation, this interface also takes Temporal dictionary contents (jp.go.nict.langrid.service_1_2.bilingualdictionary.Translation arrays) and target language in arguments. For details, please see the “Translation service combined with Temporal Dictionary” Java interface JavaDoc, which can be found in “3.2 Language Service Interface.”

Calling Language Services for Bindings

The translation combined with bilingual dictionary service and longest matching dictionary coordinated translation service are brought into realization by the workflow that appears in Appendix A. Invocations of morphological analysis, bilingual dictionaries with bilingual/longest matching search functions, and translation services are specified upon the execution of the service to be used. Bindings are optional for calling bilingual dictionaries with bilingual/longest matching search functions, but necessary for calling morphological analysis and translation services. When the dictionary coordinated service or longest matching dictionary coordinated service are executed with no bindings on morphological analysis or translation service, the service does not end normally and instead returns an exception. Please see section 3.1.1 for more details on binding functions.

The following table shows three things: call names defined by the service, classifications of services that may be bound to the corresponding call name, and conditions for supported languages.

Call Name	**Service Class**	**Supported Languages**
MorpholocicalAnalysisPL	Morphological analysis service	Must be compatible with the source language passed to the service.
TranslationPL	Translation service	Must be compatible with the source language passed to the service.
BilingualDictionaryPL **（Translation combined with bilingual dictionary services only）**	Bilingual dictionary service	Each direction word language and translation word language must be compatible with the source language passed to the service and dictTargetLang, respectively.
BilingualDictionaryWithLongestMatchSearchPL **（Longest matching dictionary coordinated translation service only）**	Bilingual dictionary service with longest matching search	Each direction word language and translation word language must be compatible with the source language passed to the service and dictTargetLang, respectively.

5.3. Utilization Methods

This section describes utilization methods for translation combined with the bilingual dictionary service and longest matching dictionary coordinated translation service.

Dynamic binding is essential in these two services, as calling the translation service without any bindings prevents the service from being executed normally, and subsequently returns an exception.

On the other hand, services for morphological analysis and bilingual dictionaries with bilingual dictionary/longest match search functions can be executed normally without bindings. However, morphological analysis with no binding means that morphological information cannot be taken into account, so even if dictionaries are bound, words from the dictionaries are not replaced, and there is a possibility that the machine translation will be output as is.

WSDL Acquisition Method for Translation Combined with Bilingual Dictionary

1. Open the View of Language Grid menu on the service manager (http://langrid.org/operation/service_manager/) and choose the Language Services page. From the Composite Services list on that page, display the Service Profile for [Translation Combined With Bilingual Dictionary] (translation combined with bilingual dictionary service) or [Translation Combined With Bilingual Dictionary With Longest Match Search] (translation combined with bilingual dictionary with longest match search service) from its respective Service Name link.
2. You may them access the service's WSDL file from the WSDL link on its Service Profile page.
3. Then, open that file in a browser and save it or right-click the link and execute the "Save As" command. **When choosing the "Save As" option, give the file the same name as the ServiceId and make sure to use a ".wsdl" extension.**

Service ID Acquisition Method for Binding Services

1. Open the View of Language Grid menu on the service manager (http://langrid.org/operation/service_manager/) and then the Language Services page.
2. Then, look for dynamically bindable services on the resulting page. See 5.2. Translation Combined with Bilingual Dictionary Specifications for more on services that can be bound via language service invocations.
3. Check whether or not the found services comply with languages passed at language service invocation by looking at the Language item.
4. Compliance with the above enables binding, so open Service Profile from the corresponding Service Name item and acquire the Service ID from the Service ID item.

Calling via Dynamic Binding

1. In accordance with programming language methods, we will now create a client module[1] to call a Web service from the acquired translation combined with bilingual dictionary / longest

[1] SoapUI (http://www.soapui.org/), provided by eviware, exists as a GUI tool for calling Web services. By inserting binding information into the SOAP request message created using this tool, you can perform call tests. For more information on usage methods, see the Services Computing School's Exercise: Basic of Web Services1 (http://langrid.org/services/jp/exercise1.html).

match translation combined with bilingual dictionary service WSDL.

2. Next, insert binding information into the SOAP header (<soapenv:Header>～ </soapenv:Header>). Please use the template for binding information that appears below. **Bolded** items are call names for binding, while ***bolded and italicized*** pieces should be replaced by IDs and URLs for services to be bound. //～ denote comments, so please delete those areas when you use the template.

```
<ns1:binding    // Binding element start tag
   soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
   soapenv:mustUnderstand="0"
   xsi:type="soapenc:string"
   xmlns:ns1="http://langrid.nict.go.jp/process/binding/tree"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
 >
[
 {    // Binding information for the morphological analysis service invocation
   &quot;children&quot;:[],
&quot;invocationName&quot;:&quot;MorphologicalAnalysisPL&quot;,
&quot;serviceId&quot;:&quot; Service ID or URL&quot; // ex.: Mecab
 }
 ,{    // Binding information for the bilingual dictionary service invocation
   &quot;children&quot;:[],
// Delete the following line for the translation combined with bilingual dictionary with longest match search service.
&quot;invocationName&quot;:&quot;BilingualDictionaryPL&quot;,
// Delete the following line for the translation combined with bilingual dictionary service.
&quot;invocationName&quot;:&quot;BilingualDictionaryWithLongestMatchSearchPL&quot;,
&quot;serviceId&quot;:&quot; Service ID or URL&quot;    // ex.: KyotoTourismDictionaryDb
 }
 ,{    // Binding information for the translation service invocation
   &quot;children&quot;:[],
   &quot;invocationName&quot;:&quot;TranslationPL&quot;,
   &quot;serviceId&quot;:&quot; Service ID or URL&quot;    // ex.: NICTJServer
 }
]
</ns1:binding>    // Binding element end tag
```

Note *See 3.1. Dynamic Binding Specification Functions for detailed information regarding dynamic bindings.*

3. Set the translation combined with bilingual dictionary / translation combined with bilingual dictionary with longest match search service arguments for the client module and execute.

The table below shows examples of argument settings.

sourceLang	Translation source language code	ja
targetLang	Translation target language code	en
source	Character string to be translated	京都の比叡山を含む東山は東山３６峰とも呼ばれています.
temporalDict	Temporal dictionary	new Translation[]{new Translation("東山３６峰", new String[]{"HIGASHIYAMA36HOU"})}
dictTargetLang	Bilingual dictionary translation language code	en

Appendix B contains SOAP messages that are transmitted and received in the above sequence.

Sample Code for Calling Translation combined with bilingual dictionary (JAVA version)

This section gives sample Java code using the translation combined with bilingual dictionary / translation combined with bilingual dictionary with longest match search service with dynamic binding.

Sample Code (SampleClient.java)

Change ***bolded and italicized*** portions accordingly.

```
package jp.go.nict.langrid.sample;
import static jp.go.nict.langrid.language.ISO639_1LanguageTags.en;
import static jp.go.nict.langrid.language.ISO639_1LanguageTags.ja;
import java.net.URL;
import java.util.Collection;
import jp.go.nict.langrid.client.ws_1_2.ClientFactory;
import jp.go.nict.langrid.client.ws_1_2.TranslationWithTemporalDictionaryClient;
import jp.go.nict.langrid.commons.cs.binding.BindingNode;
import jp.go.nict.langrid.service_1_2.bilingualdictionary.Translation;

public class SampleClient{
  public static void main(String[] args) throws Exception{
    TranslationWithTemporalDictionaryClient twtdc =
    ClientFactory.createTranslationWithTemporalDictionaryClient(
      new URL("http://langrid.nict.go.jp/langrid-1.2/invoker/"
      // Specify one of the following depending on the service to be called.
      + "TranslationCombinedWithBilingualDictionary")
      //+ "TranslationCombinedWithBilingualDictionaryWithLongestMatchSearch")
    );
```

```
    twtdc.setUserId("Language grid ID");
    twtdc.setPassword("password");
    Collection<BindingNode> bindings = twtdc.getTreeBindings();
    bindings.add(new BindingNode("MorphologicalAnalysisPL", "Mecab"));
    bindings.add(new BindingNode("TranslationPL", "NICTJServer"));
    bindings.add(
    // When using a bilingual dictionary, specify the appropriate item from the following in accordance
    // with the translation combined with    bilingual dictionary service / translation combined with
    // bilingual dictionary with longest match search service.
        new BindingNode("BilingualDictionaryPL",
        //new BindingNode("BilingualDictionaryWithLongestMatchSearchPL",
            "KyotoTourismDictionaryDb"));
    String message = "京都の比叡山を含む東山は東山３６峰とも呼ばれています.  ";
    System.out.println(message);
    String result = twtdc.translate(
     ja, en
      , message,
        new Translation[]{
          new Translation("東山３６峰", new String[]{"HIGASHIYAMA36HOU"})
        }
        // When not using a temporal dictionary, pass a null array as follows:
        //, new Translation[]{}
        , en);
    System.out.println(result);
  }
}
```

6. Contact Information

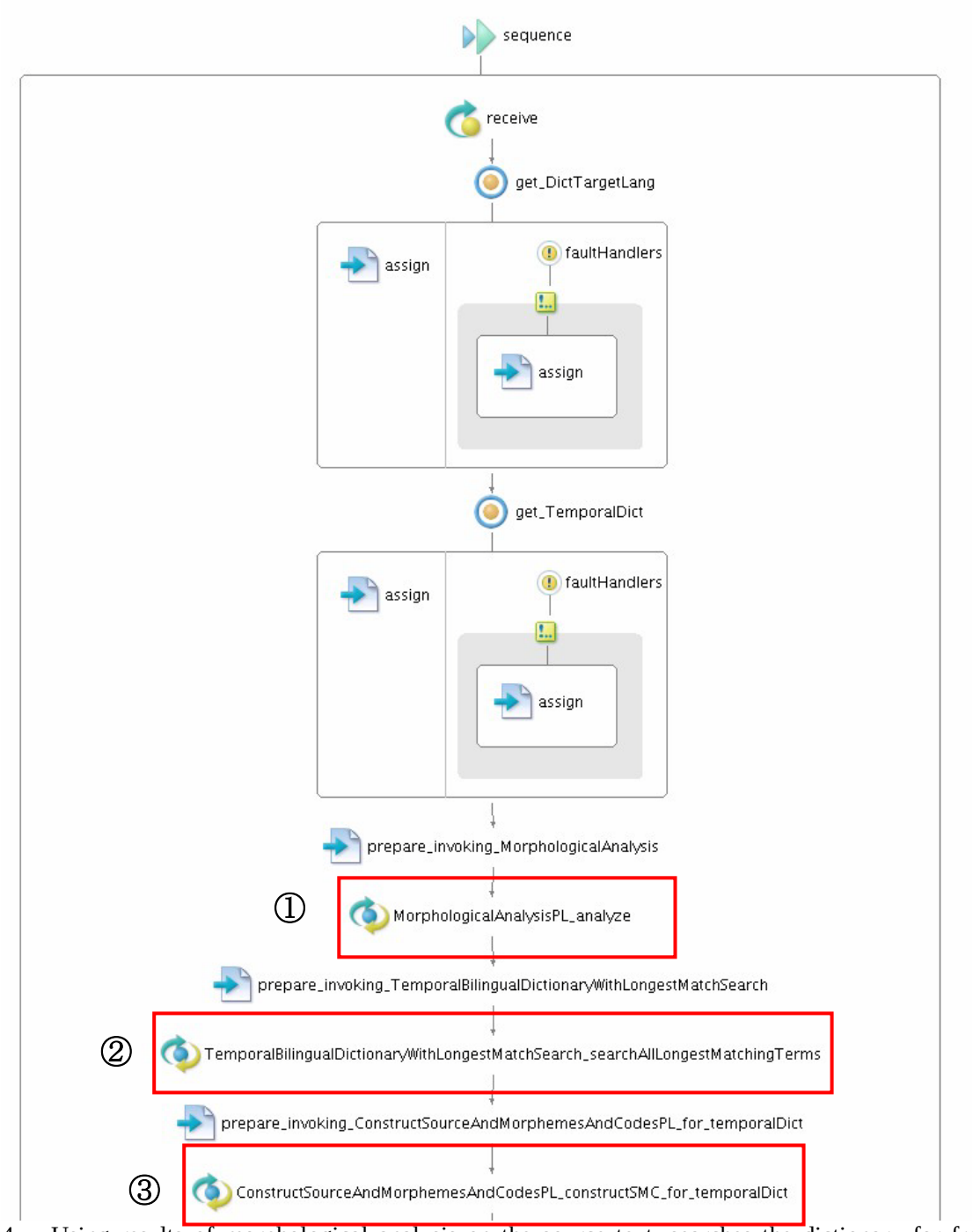
Should you have any questions about this manual, please contact the following address.

Language Grid Support Section: langrid@khn.nict.go.jp

Appendix A：Workflow Explanation

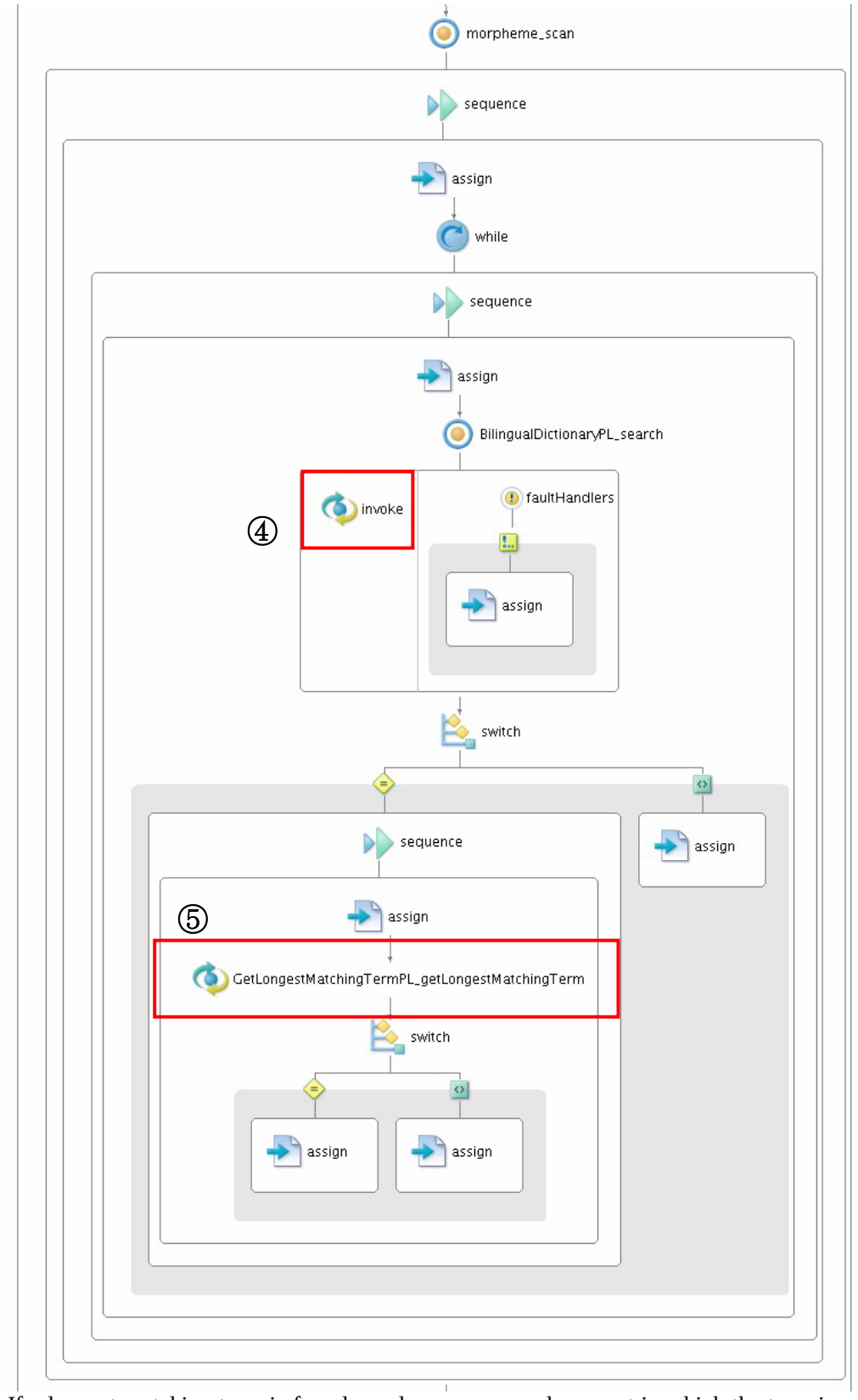
Below is a work flow chart for the translation combined with bilingual dictionary service.

1. Calls the morphological analyses service and acquires morphological analysis results from the source document.
2. Searches for the longest-matching terms that match source document terms from the Temporal dictionary's direction words.
3. If a longest matching term is found, produces a source document in which the term is replaced with intermediate code (insignificant character string that does not affect translation).



4. Using results of morphological analysis on the source text, searches the dictionary for front matches, beginning with the first morpheme.
5. Acquires terms from the dictionary search results' direction words that satisfy longest matching

conditions with the source text.



6. If a longest matching term is found, produces a source document in which the term is replaced with intermediate code.

7. Translates the source text containing intermediate code.
8. Replaces intermediate code contained in the translation result with translations from Temporal dictionaries and/or bilingual dictionaries.

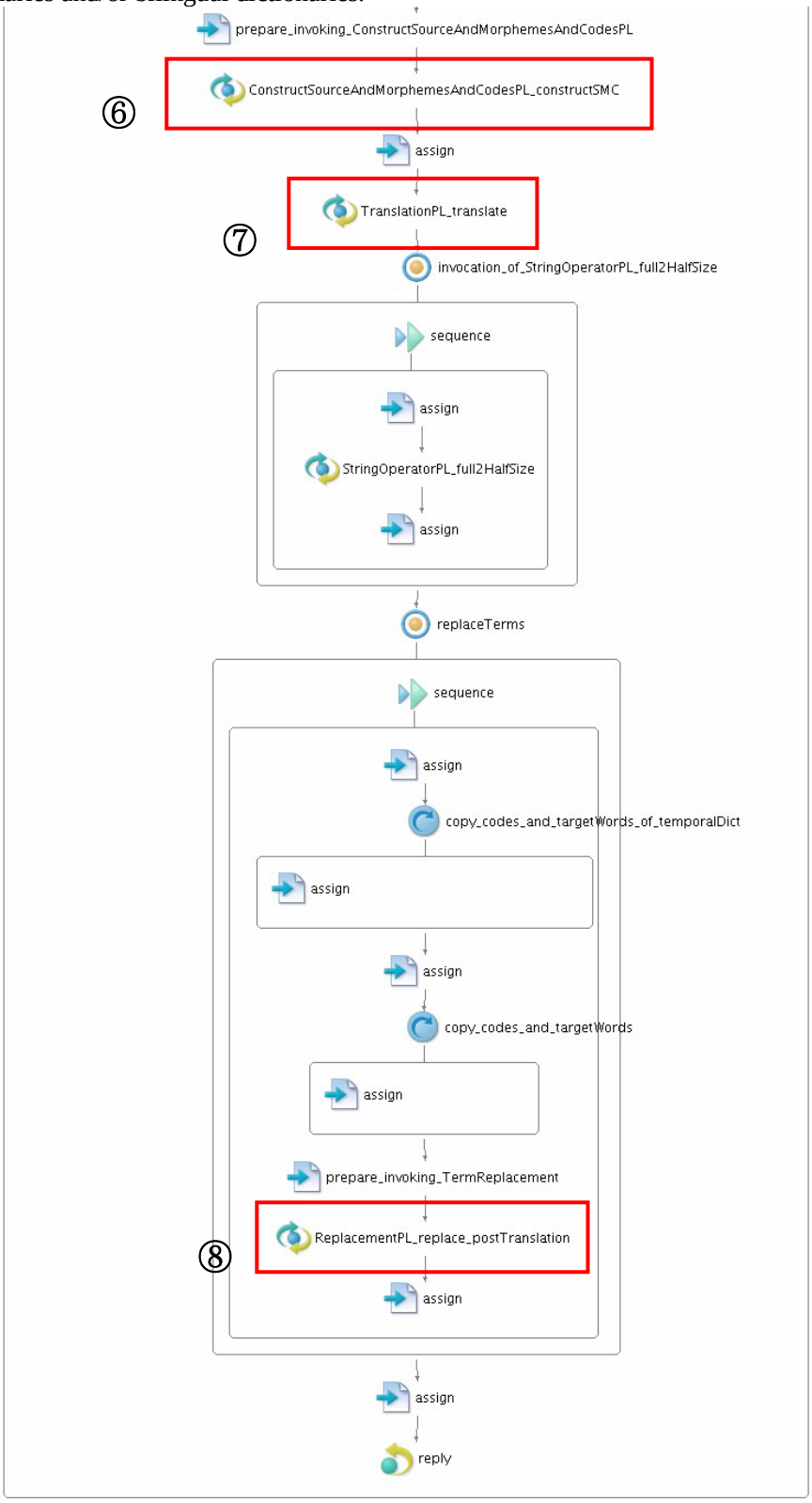
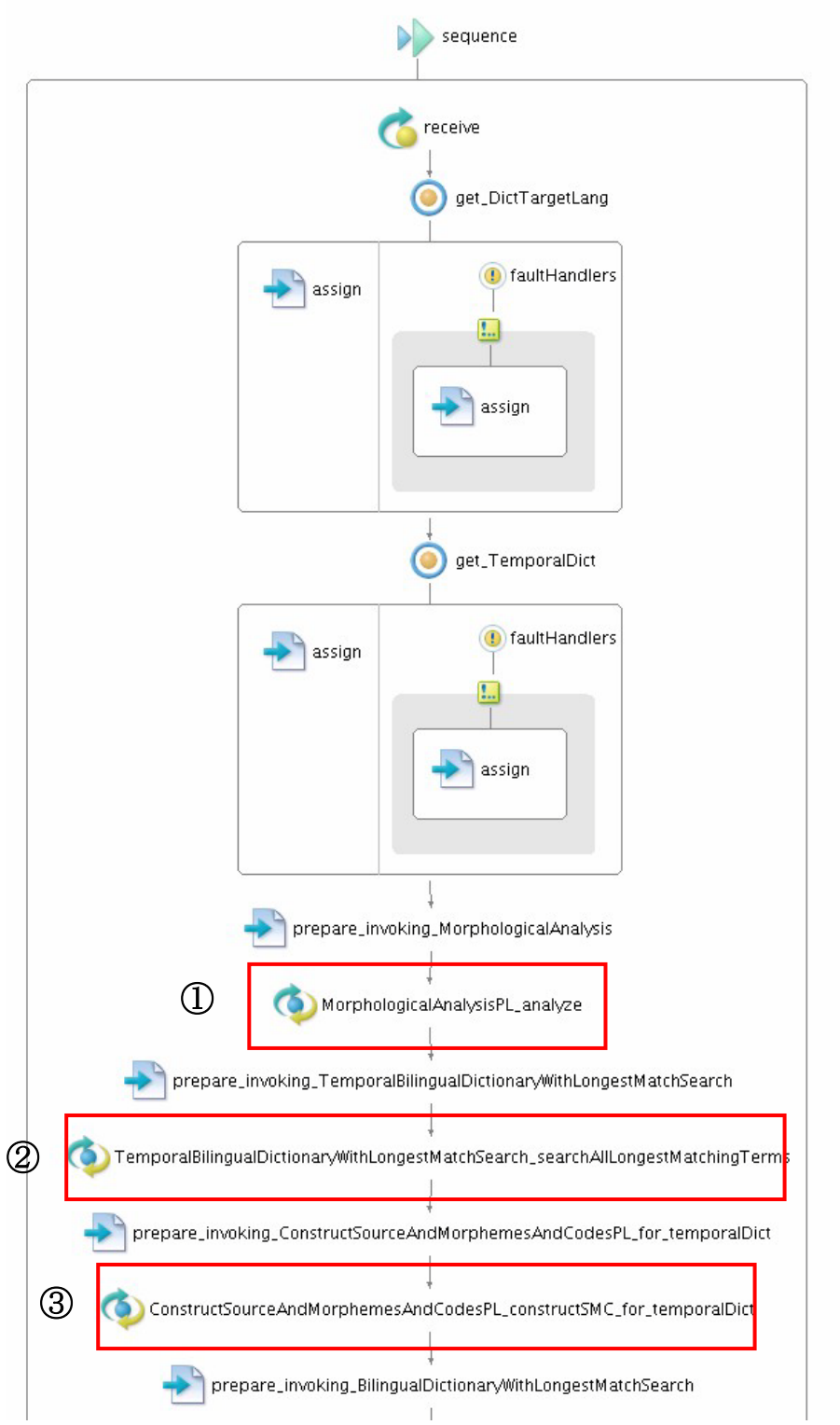


Figure 2 shows a workflow for longest matching dictionary coordinated translation.

1. Calls the morphological analysis service and acquires morphological analysis results from the source

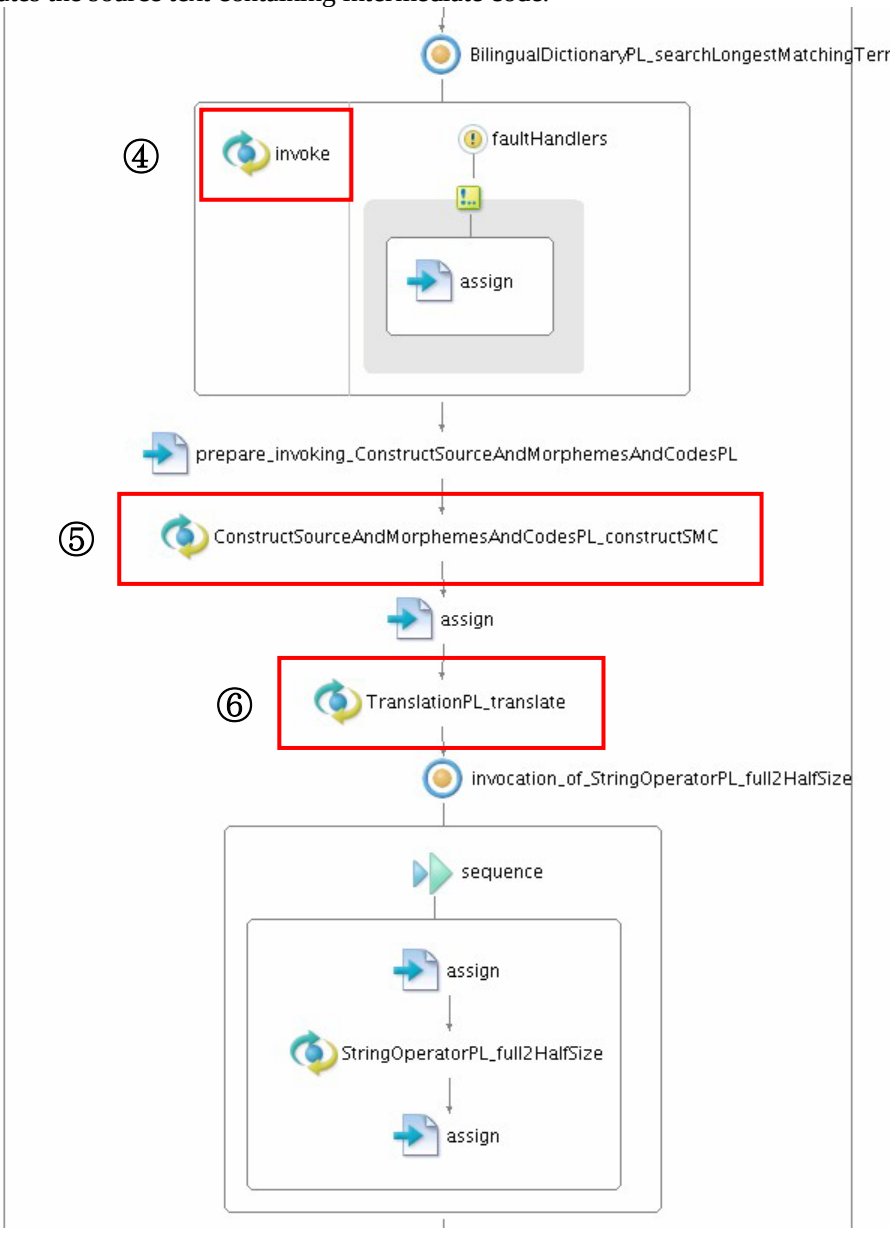
text.

2. Searches for the longest-matching terms that match source document terms from the Temporal dictionary's direction words.
3. If a longest matching term is found, produces a source document in which the term is replaced with intermediate code.

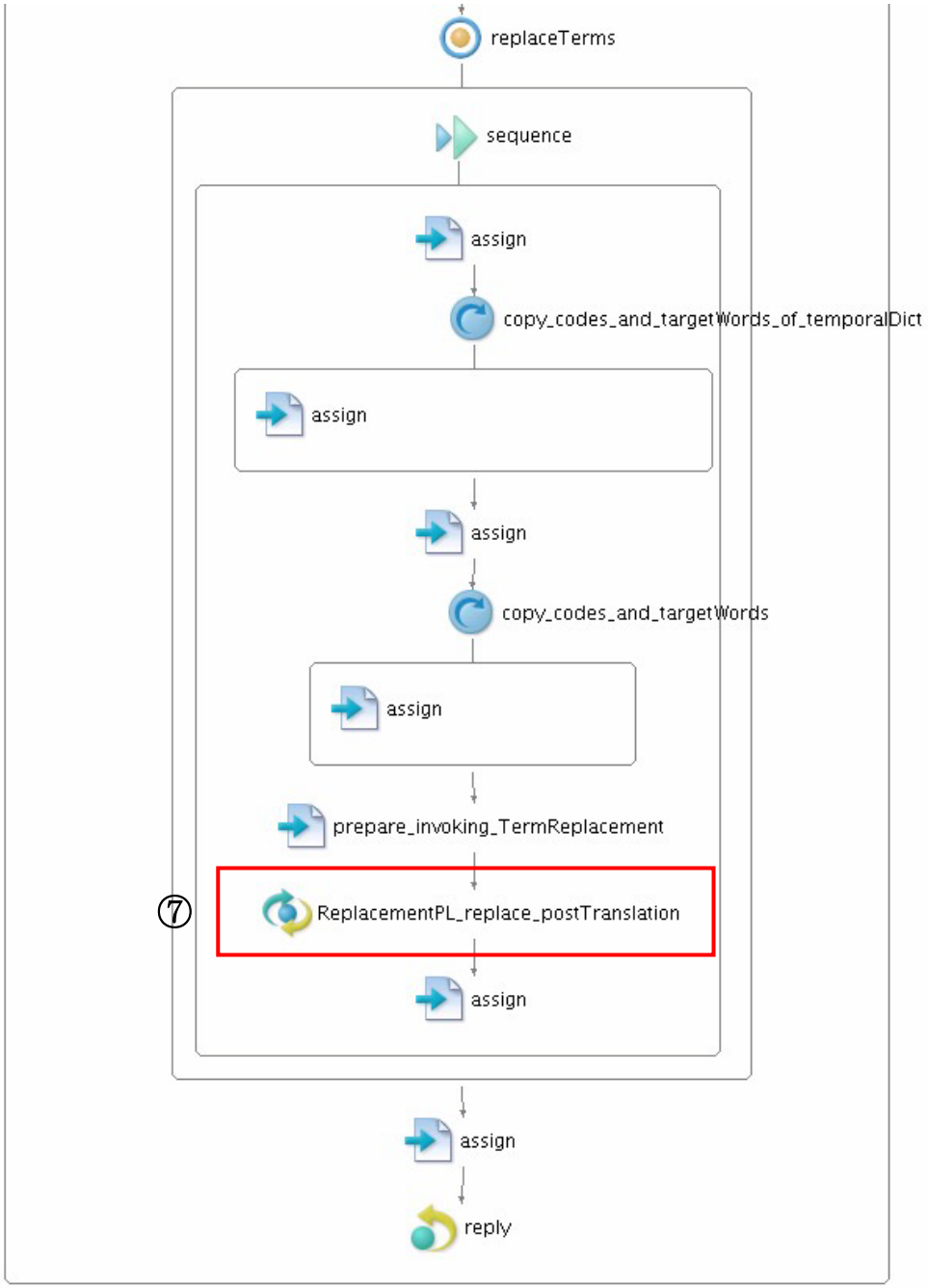


4. Searches for the longest-matching terms that match source text terms from the bilingual dictionary's direction words.

5. If a longest matching term is found, produces a source document in which the term is replaced with intermediate code.
6. Translates the source text containing intermediate code.



7. Replaces intermediate code contained in the translation result with translations from Temporal dictionaries and/or bilingual dictionaries.


replaceTerms
sequence
assign
copy_codes_and_targetWords_of_temporalDict
assign
assign
copy_codes_and_targetWords
assign
prepare_invoking_TermReplacement
⑦
ReplacementPL_replace_postTranslation
assign
assign
reply

Appendix B： SOAP Messages

Below, find examples of SOAP messages transmitted and received when the arguments indicated in the manual are set and the translation combined with bilingual dictionary / translation combined with bilingual dictionary with longest match search service are invoked. Each argument is stored within the SOAP body, while dynamic binding is specified in the SOAP header. HTML is escaped.

＜Request Message＞

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
<soapenv:Header>
 <ns1:binding
  soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
  soapenv:mustUnderstand="0"
  xsi:type="soapenc:string"
  xmlns:ns1="http://langrid.nict.go.jp/process/binding/tree"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
 >
 [
  {
   &quot;children&quot;:[],
   &quot;invocationName&quot;:&quot;MorphologicalAnalysisPL&quot;,
   &quot;serviceId&quot;:&quot;Mecab&quot;
  }
  ,{
   &quot;children&quot;:[],
   //Specify one of the following based on the type of translation with bilingual dictionary service.
   &quot;invocationName&quot;:&quot;BilingualDictionaryPL&quot;,
   //&quot;invocationName&quot;:&quot;BilingualDictionaryWithLongestMatchSearchPL&quot;,

   &quot;serviceId&quot;:&quot;KyotoTourismDictionaryDb&quot;
  }
  ,{
   &quot;children&quot;:[],
   &quot;invocationName&quot;:&quot;TranslationPL&quot;,
   &quot;serviceId&quot;:&quot;NICTJServer&quot;
  }
 ]
 </ns1:binding>
</soapenv:Header>
<soapenv:Body>
  <translate soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> // Method name
   <sourceLang xsi:type="xsd:string">ja</sourceLang> // Translation source language
```

```
    <targetLang xsi:type="xsd:string">en</targetLang> // Translation target language
    <source xsi:type="xsd:string"> // Source text
      &#x4EAC;&#x90FD;&#x306E;&#x6BD4;&#x53E1;&#x5C71;&#x3092;&#x542B;&#x3080;&#
x6771;&#x5C71;&#x306F;&#x6771;&#x5C71;&#xFF13;&#xFF16;&#x5CF0;&#x3068;&#x3082;&#x
547C;&#x3070;&#x308C;&#x3066;&#x3044;&#x307E;&#x3059;&#x3002;
    </source>
    <temporalDict    // Temporal dictionary
     soapenc:arrayType="ns2:Translation[1]" // Number of translations specified by the Temporal
dictionary
     xsi:type="soapenc:Array" xmlns:ns2="http://langrid.nict.go.jp/ws_1_2/bilingualdictionary/"
     xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    >
     <temporalDict href="#id0"/> // Reference to a Temporal dictionary element
    </temporalDict>
    // Elements will be empty, as shown below, unless a Temporal dictionary is specified.
    // <temporalDict
    //    soapenc:arrayType="ns2:Translation[0]" // Empty, so "0" is specified.
    //    xsi:type="soapenc:Array"
    //    xmlns:ns2="http://langrid.nict.go.jp/ws_1_2/bilingualdictionary/"
    //    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    //    />
    <dictTargetLang xsi:type="xsd:string">en</dictTargetLang> // Bilingual dictionary translation
language
   </translate>
   <multiRef id="id0" // 1 Temporal dictionary element
    soapenc:root="0"
    soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    xsi:type="ns3:Translation"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:ns3="http://langrid.nict.go.jp/ws_1_2/bilingualdictionary/"
   >
    <headWord xsi:type="xsd:string"> // Temporal dictionary direction word
     &#x6771;&#x5C71;&#xFF13;&#xFF16;&#x5CF0;
    </headWord>
    <targetWords
     soapenc:arrayType="xsd:string[1]" // Number of words compliant with the direction word
     xsi:type="soapenc:Array"
    >
     <targetWords xsi:type="xsd:string"> // Temporal dictionary translation
      HIGASHIYAMA36HOU
     </targetWords>
    </targetWords>
   </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP response messages received from the translation combined with bilingual dictionary / longest match dictionary coordinate language service are as follows. Call trees (execution logs) for called services are stored in the SOAP header, and translation results in the SOAP body. In addition, like requests, HTML is escaped.

Note *See 3.1.3. Call Tree Return Functions for detailed information regarding call trees.*

＜Translation Combined with Bilingual Dictionary Service Response Message＞

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
 <soapenv:Header>
  <ns1:calltree
   soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
   soapenv:mustUnderstand="0"
   xsi:type="soapenc:string"
   xmlns:ns1="http://langrid.nict.go.jp/process/calltree"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  >
   [
    {
     &quot;children&quot;:[]
     ,&quot;faultCode&quot;:&quot;&quot; // Fault code at failure
     ,&quot;faultString&quot;:&quot;&quot; // Fault string at failure
     ,&quot;responseTimeMillis&quot;:127 // Execution time of this service
     ,&quot;serviceCopyright&quot;:&quot;Taku Kudo, and Nippon Telegraph and Telephone
Corporation&quot; // Copyright for this service
     ,&quot;serviceId&quot;:&quot;Mecab&quot; // ID of this service
     ,&quot;serviceLicense&quot;:&quot;http://mecab.sourceforge.net/&quot; // License for this
service
     ,&quot;serviceName&quot;:&quot;MeCab&quot; // Name of this service
    }
    ,{
     &quot;children&quot;:[]
     ,&quot;faultCode&quot;:&quot;&quot;
     ,&quot;faultString&quot;:&quot;&quot;
     ,&quot;responseTimeMillis&quot;:179
     ,&quot;serviceCopyright&quot;:&quot;&quot;
     ,&quot;serviceId&quot;:&quot;TemporalBilingualDictionaryWithLongestMatchSearch&quot;
     ,&quot;serviceLicense&quot;:&quot;&quot;
     ,&quot;serviceName&quot;:&quot;Temporal Bilingual Dictionary With Longest Match
Search&quot;
```

```
      }
      ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
        ,&quot;responseTimeMillis&quot;:171
        ,&quot;serviceCopyright&quot;:&quot;&quot;
        ,&quot;serviceId&quot;:&quot;ConstructSourceAndMorphemesAndCodes&quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;Construct Source And Morphemes And Codes&quot;
      }
      ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
        ,&quot;responseTimeMillis&quot;:12
        ,&quot;serviceCopyright&quot;:&quot;&quot;
        ,&quot;serviceId&quot;:&quot;KyotoTourismDictionaryDb&quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;&quot;
      }
      ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
        ,&quot;responseTimeMillis&quot;:275
        ,&quot;serviceCopyright&quot;:&quot;&quot;
        ,&quot;serviceId&quot;:&quot;GetLongestMatchingTerm&quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;Get Longest Matching Term&quot;
      }
-Repeat the same lines-
      ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
        ,&quot;responseTimeMillis&quot;:144
        ,&quot;serviceCopyright&quot;:&quot;&quot;
        ,&quot;serviceId&quot;:&quot;ConstructSourceAndMorphemesAndCodes&quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;Construct Source And Morphemes And Codes&quot;
      }
      ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
```

```
        ,&quot;responseTimeMillis&quot;:46
        ,&quot;serviceCopyright&quot;:&quot;Kodensha Co., Ltd.&quot;
        ,&quot;serviceId&quot;:&quot;NICTJServer&quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;J-Server&quot;
       }
       ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
        ,&quot;responseTimeMillis&quot;:3
        ,&quot;serviceCopyright&quot;:&quot;&quot;
        ,&quot;serviceId&quot;:&quot;http://192.168.0.2:8080/collabo/services/StringOperationService&
quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;&quot;
       }
       ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
        ,&quot;responseTimeMillis&quot;:85
        ,&quot;serviceCopyright&quot;:&quot;&quot;
        ,&quot;serviceId&quot;:&quot;TermReplacementService&quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;Term Replacement Service&quot;
       }
     ]
    </ns1:calltree>
  </soapenv:Header>
  <soapenv:Body>
    <ns2:translateResponse
     soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
     xmlns:ns2="http://translation.ws_1_2.wrapper.langrid.nict.go.jp"
    >
     <translateReturn xsi:type="xsd:string"> // Translation result
       Higashiyama including Mt. Hiei in Kyoto is also called HIGASHIYAMA36HOU .
     </translateReturn>
    </ns2:translateResponse>
  </soapenv:Body>
</soapenv:Envelope>
```


＜Translation Combined with Bilingual Dictionary with Longest Match Search Response Message＞


```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
 <soapenv:Header>
  <ns1:calltree
   soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
   soapenv:mustUnderstand="0"
   xsi:type="soapenc:string"
   xmlns:ns1="http://langrid.nict.go.jp/process/calltree"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  >
   [
    {
     &quot;children&quot;:[]
     ,&quot;faultCode&quot;:&quot;&quot; // Fault code at failure
     ,&quot;faultString&quot;:&quot;&quot; // Fault string at failure
     ,&quot;responseTimeMillis&quot;:70 // Execution time of this service
     ,&quot;serviceCopyright&quot;:&quot;Taku Kudo, and Nippon Telegraph and Telephone
Corporation&quot; // Copyright for this service
     ,&quot;serviceId&quot;:&quot;Mecab&quot; // ID of this service
     ,&quot;serviceLicense&quot;:&quot;http://mecab.sourceforge.net/&quot; // License for this
service
     ,&quot;serviceName&quot;:&quot;MeCab&quot; // Name of this service
    }
    ,{
     &quot;children&quot;:[]
     ,&quot;faultCode&quot;:&quot;&quot;
     ,&quot;faultString&quot;:&quot;&quot;
     ,&quot;responseTimeMillis&quot;:146
     ,&quot;serviceCopyright&quot;:&quot;&quot;
     ,&quot;serviceId&quot;:&quot;TemporalBilingualDictionaryWithLongestMatchSearch&quot;
     ,&quot;serviceLicense&quot;:&quot;&quot;
     ,&quot;serviceName&quot;:&quot;Temporal Bilingual Dictionary With Longest Match
Search&quot;
    }
    ,{
     &quot;children&quot;:[]
     ,&quot;faultCode&quot;:&quot;&quot;
     ,&quot;faultString&quot;:&quot;&quot;
     ,&quot;responseTimeMillis&quot;:172
     ,&quot;serviceCopyright&quot;:&quot;&quot;
     ,&quot;serviceId&quot;:&quot;ConstructSourceAndMorphemesAndCodes&quot;
     ,&quot;serviceLicense&quot;:&quot;&quot;
     ,&quot;serviceName&quot;:&quot;Construct Source And Morphemes And Codes&quot;
```

```
      }
      ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
        ,&quot;responseTimeMillis&quot;:100
        ,&quot;serviceCopyright&quot;:&quot;&quot;
        ,&quot;serviceId&quot;:&quot;KyotoTourismDictionaryDb&quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;&quot;
      }
      ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
        ,&quot;responseTimeMillis&quot;:142
        ,&quot;serviceCopyright&quot;:&quot;&quot;
        ,&quot;serviceId&quot;:&quot;ConstructSourceAndMorphemesAndCodes&quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;Construct Source And Morphemes And Codes&quot;
      }
      ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
        ,&quot;responseTimeMillis&quot;:64
        ,&quot;serviceCopyright&quot;:&quot;Kodensha Co., Ltd.&quot;
        ,&quot;serviceId&quot;:&quot;NICTJServer&quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;J-Server&quot;
      }
      ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
        ,&quot;responseTimeMillis&quot;:3
        ,&quot;serviceCopyright&quot;:&quot;&quot;
        ,&quot;serviceId&quot;:&quot;http://192.168.0.2:8080/collabo/services/StringOperationService&
quot;
        ,&quot;serviceLicense&quot;:&quot;&quot;
        ,&quot;serviceName&quot;:&quot;&quot;
      }
      ,{
        &quot;children&quot;:[]
        ,&quot;faultCode&quot;:&quot;&quot;
        ,&quot;faultString&quot;:&quot;&quot;
```

```
        ,&amp;quot;responseTimeMillis&amp;quot;:18
        ,&amp;quot;serviceCopyright&amp;quot;:&amp;quot;&amp;quot;
        ,&amp;quot;serviceId&amp;quot;:&amp;quot;TermReplacementService&amp;quot;
        ,&amp;quot;serviceLicense&amp;quot;:&amp;quot;&amp;quot;
        ,&amp;quot;serviceName&amp;quot;:&amp;quot;Term Replacement Service&amp;quot;
       }
     ]
   </ns1:calltree>
  </soapenv:Header>
  <soapenv:Body>
   <ns2:translateResponse
    soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:ns2="http://translation.ws_1_2.wrapper.langrid.nict.go.jp"
   >
    <translateReturn xsi:type="xsd:string"> //Translation result
     Higashiyama including Mt. Hiei in Kyoto is also called HIGASHIYAMA36HOU .
    </translateReturn>
   </ns2:translateResponse>
  </soapenv:Body>
</soapenv:Envelope>
```