

## A Layered Language Service Architecture for Intercultural Collaboration

Yohei Murakami<sup>1</sup> and Toru Ishida<sup>1,2</sup>

<sup>1</sup>*Language Grid Project, National Institute of Information and Communications Technology,  
3-5 Hikaridai, Seika-cho, Kyoto, Japan*

<sup>2</sup>*Department of Social Informatics, Kyoto University,  
Yoshida-honmachi, Kyoto, Japan  
yohei@nict.go.jp, ishida@i.kyoto-u.ac.jp*

### Abstract

*The Language Grid is an infrastructure for enabling users to share language resources developed by language specialists and end-user communities. The users employ the resources to support their intercultural/multilingual activities. However, there is a big gap between functions provided by the language resources and those needed by the users. To bridge the gap and satisfy users' needs, this paper extends the Language Grid by adding four language service layers: system adaptation layer, combination layer, coordination layer, and user adaptation layer. By introducing the four-layers, the Language Grid can coordinate several language resources to create new language services customized to users' activities, such as back translations, multi-hop translations, and specialized translations.*

### 1. Introduction

Though there are many language resources (both data and programs) on the Internet [2], many ongoing intercultural collaboration activities are still lacking multilingual support. For example, some cities provide information of administrative services in some foreign languages on the Web, while other cities cannot provide any multilingual support for foreigners because they cannot reuse the information due to copyright.

To solve such problems, the Language Grid has been proposed as a new infrastructure on the Web; it allows the sharing of language resources developed by language specialists and end-user communities [6]. Moreover, the Language Grid is being operated on a non-profit operation model to match the incentives of stakeholders: Language Resource Providers,

Computation Resource Providers, Language Service Users, and the Language Grid Operator [8]. As a result, the Language Grid successfully improves the accessibility of language resources. The Language Grid Association has been organized as a user group to discuss and accumulate knowledge to better utilize the Language Grid from various perspectives [11], and allow end-users to employ language resources in their daily life [7].

The above does not mean that the Language Grid has radically increased the usability of language resources. The language resources remain difficult to use. For example, one NPO<sup>1</sup> has been creating a “universal playground” for kids around the world. To plan activities in Japan, Korea, Kenya and Austria, volunteer facilitators need a translation service with multilingual glossary in order to translate sentences including their community words. Another NPO<sup>2</sup> assisting foreign patients needs to support health interview by combining conversation corpus in medical domain and parallel texts.

These demands cannot be satisfied with interfaces of language resources that are defined in language resource ontology [4][5]. That is, we cannot respond to all the demands with a single language resource on the Language Grid. This is because language resources have been developed in process of researching of natural language processing and linguistics, not to satisfy application side demands. Therefore, we have to construct an infrastructure that not only increases the accessibility of language resources but also increases their usability by combining original language resources.

---

<sup>1</sup> NPO Pangaea. <http://www.pangaeaan.org/>.

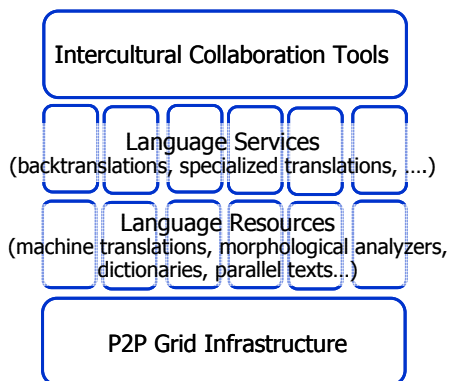
<sup>2</sup> Kyoto Center for Multicultural Society  
<http://www.tabunka.jp/kyoto/>.

In this paper, we construct a language service layer between language resources and intercultural collaboration tools in order to produce composite language services. The language service layer contains various types of Web services that are useful for Language Grid applications including intercultural collaboration tools. Moreover, we have made it efficient to create new language services by further dividing the language service layer into four layers, because as the number of Language Grid applications increases, more language services are needed. Besides the four-layered language service architecture, we show how to integrate similar language services based on different language resources into one service by dynamically selecting the language resources invoked.

The rest of paper describes the service layer of the Language Grid, the four-layered language services, and the language grid service manager, a web-based tool to control and monitor the behavior of language services.

## 2. Service layer of the language grid

As shown in Figure 1, the Language Grid consists of four service layers.



**Figure 1. Service layer**

- a) Intercultural Collaboration Tools  
The interface at the top layer provides Intercultural Collaboration Tools so that the users can utilize the Language Grid, even if they have no programming skills. Collaboration tools are developed using language services explained below. Though the Language Grid provides several tools, new tools can be developed by Language Service Users, and existing tools can be multilingualized.
- b) Language Services

Various language services for intercultural collaboration can be created by combining existing language resources. We have already implemented Web service workflows including *back translations*<sup>3</sup> and *domain-specialized translations*. Language Service Users can easily add new language services to the Language Grid by themselves.

- c) Language Resources  
Various language resources will be provided as atomic services with a standardized interface. To increase usability of the language resources for language services, standardization of access entry is quite important [1]. We started working on language resource ontology, which standardizes the interfaces [4][5]. Language Resource Providers can easily add new language resources to the Language Grid, and access the usage statistics of their resources.
- d) P2P Grid Infrastructure  
This infrastructure organizes multiple servers, *language grid core nodes* and *language grid service nodes* [10], on the Internet to satisfy end users' requests. Computation Resource Providers can add their servers to the P2P Grid, and access the usage statistics of their resources.

## 3. Language services

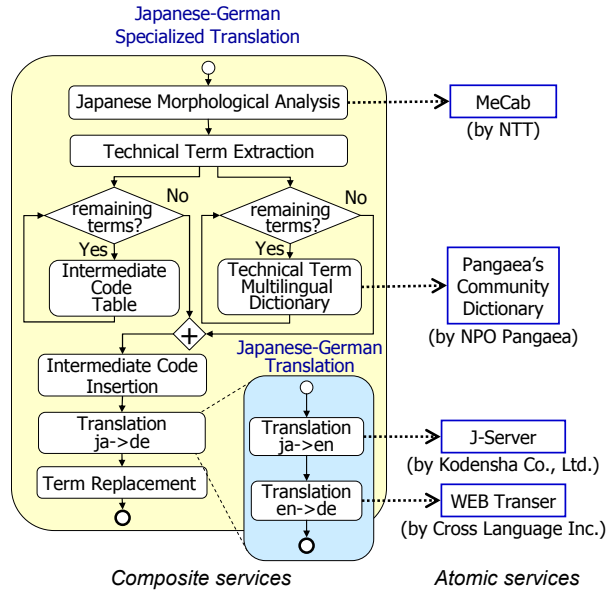
Among existing research, EuroWordNet [12] and Global WordNet Grid [3] are pioneer works on connecting dictionaries in different languages based on word semantics. The Language Grid, however, is a trial to build an infrastructure that can combine various language resources based on workflows.

The Language Grid uses BPEL4WS [9] to describe workflows, and uses the BPEL engine to execute them; the engine invokes Web services as specified in the workflow. The following two types of language services are available in the Language Grid:

- Atomic service:  
A Web service that corresponds to an individual language resource. Examples include Bilingual dictionaries, parallel texts, morphological analyzers, and machine translators.
- Composite service:  
An advanced service described by a workflow that combines several atomic services. Examples include domain-specialized translations, and multilingual back translations.

<sup>3</sup> To allow users to check the accuracy of translation, intercultural collaboration tools often provide *back translation*, which translates the translated sentences into the original input language.

Figure 2 shows a typical composite service: several atomic services and composite services are combined to create a domain specialized translation service. Domain specialized translation workflow invokes several atomic services, MeCab (Japanese morphological analyzer) and Pangaea Community Dictionary (multilingual dictionary), and a two-hop translation service consisting of J-Server (machine translator) and WEB-Transter (machine translator).



**Figure 2. Example of composite service**

To increase the flexibility of language service composition, a wide variety of language resources must be registered as atomic services. However, available language resources are often limited. There are two available machine translators in the current Language Grid. English is often the hub of translation, and that makes back translation complex. If we use Japanese-German back translation, we have to combine four translators provided by different organizations. We are observing how the quality of translation impacts communication effectiveness [13]. The findings will contribute to developing new technologies to coordinate multiple machine translators.

Our initial observations show that the Language Grid needs to allow end-users to create their own community dictionaries in order to enhance domain-specific language resources for practical intercultural collaboration support. Moreover, workflows to coordinate the language resources need to be classified according to users' needs, and the dependency between the workflows needs to be specified. This will

enable users to choose language resources and language services to construct concrete workflows. Furthermore, we must develop a dynamic binding function on the existing workflow execution engine that can associate language resources and services to workflows at runtime. This function will improve the performance of the engine by decreasing workflows statically deployed on the engine

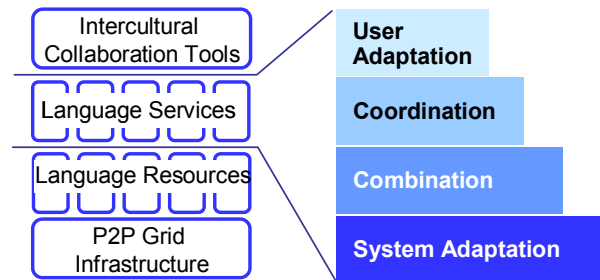
### 3.1. Community language resource creation

Community language resource includes technical terms and examples used within a community. The Language Grid provides services to multilingualize these resources in order to enhance language resources.

Creating community language resources is different from individually wrapping existing language resources to support various data formats. By creating language resources with a pre-defined common data format, a single wrapper can be reused. This enables users to register atomic services to the Language Grid without regard to wrappers. In addition, search from multiple dictionaries is easily realized by sending the same query to the multiple tables because the dictionaries are stored with the common data format.

### 3.2. Four-layered language service architecture

Figure 3 illustrates our four-layered language service architecture, which is layered according to the users' needs and dependencies between workflows. Workflows in an upper layer can invoke language services corresponding to the workflows in the lower layers. Workflows in the lowest layer, system adaptation layer, can invoke only atomic services corresponding to language resources. Table 1 summarizes representative workflows in each layer.



**Figure3. Four-layered language service architecture**

**Table1. Comparison among workflows in each layer**

Layer	Workflow name	Bindings	Explanation
User Adaptation	Specialized translation	<ul style="list-style-type: none"> <li>– Translation services in the coordination layer</li> <li>– Morphological analyzer</li> <li>– Multilingual technical term dictionary</li> </ul>	By coordinating multilingual glossaries and a translation service, this workflow can translate sentences including technical terms that the translation service cannot. At first, it gets a set of morphemes using morphological analyzer. Next, it extracts a set of technical terms from an input sentence by using the set of morphemes, and gets translations of the terms from the multilingual glossaries. After that, it replaces the technical terms in the input sentence with intermediate codes, and translates the result. Finally, it replaces intermediate codes in the translation result with the correct translation.
Coordination	Coordinated translation	<ul style="list-style-type: none"> <li>– Translation services in the combination layer</li> <li>– Multilingual synonym Sets</li> </ul>	In this workflow, each translation service informs the subsequent translation services of the concepts used in the translation to solve the translation drift problem and guarantee translation consistency.
Combination	Multi-hop translation	<ul style="list-style-type: none"> <li>– Translation services in the system adaptation layer</li> </ul>	By combining multiple translation services sequentially, this workflow can translate sentences that a single translation service in the system adaptation layer cannot translate directly.
System Adaptation	Translation with pre-processing of abbreviations	<ul style="list-style-type: none"> <li>– Machine translator</li> <li>– Dictionaries of abbreviations</li> </ul>	By replacing abbreviations in an input sentence with standard nomenclature, this workflow can adapt the input sentence to machine translators in order to improve the translatability of the sentence.

**3.2.1. System adaptation layer.** The purpose of the system adaptation layer is to adapt input sentences to machine translators by improving the sentence to increase its translatability. For example, by replacing abbreviations with standard nomenclature, or by paraphrasing input sentences in plain language as pre-processing operation, workflows in this layer improve translation quality.

In the system adaptation layer, users can bind paraphrase engines, dictionaries of abbreviations, and machine translators to workflows.

**3.2.2. Combination layer.** The purpose of the combination layer is to combine multiple translation services in order to translate between language pairs that no single translation service currently supports. For example, a workflow realizes Japanese-German translation by combining Japanese-English translation service and English-German translation service sequentially. Moreover, another workflow realizes Japanese-German backtranslation by combining the Japanese-German translation service and the German-Japanese translation service. Furthermore, by combining the Japanese-English translation service, Japanese-Chinese translation service, and the Japanese-Korean translation service concurrently, another workflow realizes a service that returns a set of English, Chinese, and Korean translation results.

In the combination layer, users can bind translation services in the system adaptation layer or machine translators to workflows.

**3.2.3. Coordination layer.** The purpose of the coordination layer is to coordinate multiple translation services and multilingual synonym sets in order to solve asymmetry and inconsistency problems [13] which arise when combining multiple translation services. For example, some workflows solve these problems by sharing the context of translation selection with other translation services, or by maintaining the context of translation selection in the previous translation.

In the coordination layer, users can bind translation services in the combination layer or machine translators, and multilingual synonym sets such as EuroWordNet and Global WordNet Grid to workflows.

**3.2.4. User adaptation layer.** The purpose of user adaptation layer is to adapt output sentences to users by coordinating community language resources and translation services. For example, a workflow can translate sentences including technical terms by coordinating a multilingual community dictionary and translation services. Moreover, another workflow can translate sentences into users' dialect by coordinating translation services and a paraphrasing dictionary between the mother language and the desired dialect.

Furthermore, another workflow can translate fixed phrases within a community by coordinating translation services and parallel texts for the community.

In the user adaptation layer, users can bind translation services in the coordination layer or machine translators, multilingual glossaries, multilingual parallel texts, and morphological analyzers to workflows.

### 3.3. Dynamic web service binding

The existing workflow execution engine needs workflows that are bound language resources and services to. These workflows have to be deployed on the engine before execution. However, as the number of community language resources increases, the more language services based on the same workflow must be deployed on the engine. This decreases the performance of the engine. Therefore, we have implemented a dynamic Web service binding function in the language service layer by improving the existing workflow execution engine. The dynamic Web service binding function binds the language resources and services to a workflow at runtime according to users' requests. This function enables users to create a new language services by choosing a deployed workflow and service IDs for service binding, without knowing how to describe a workflow or how to deploy the workflow.

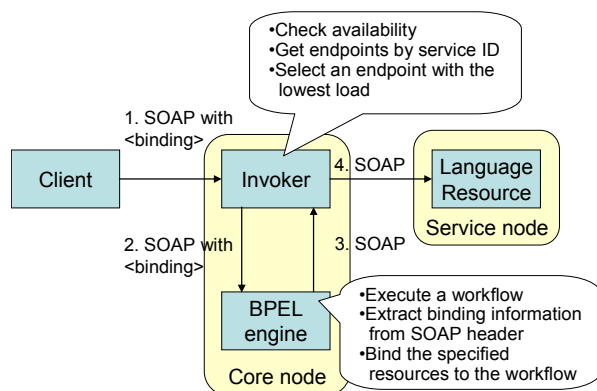


Figure 4. Dynamic web service binding

The dynamic web service binding function needs abstract workflows consisting of services with endpoints and ones without endpoints. As shown in Figure 4, the workflow execution engine confirms

whether each service in a workflow has endpoints or not at runtime by using service profiles stored in the P2P Grid infrastructure. If a service does not have any endpoints, the workflow execution engine extracts binding information, the service URI and an alternative service ID, from <binding> tags in the SOAP header. Figure 5 illustrates an example of a <binding> tag in an SOAP header. The extracted service ID is sent to the service invoker. The service invoker checks availability of the service, gets a list of endpoints to invoke the service, and then selects the endpoint with the lowest load

```

<?xml version = "1.0"?>
<soapenv:Envelope>
<soapenv:Header>
<binding xmlns=http://langrid.nict.go.jp/process/binding>
http%3a%2f%2flanres%2enict%2ego%2ejp%2flangrid%2fservices%2fJServer,CLWT
</binding>
</soapenv:Header>
<soapenv:Body>
...
</soapenv:Body>
</soapenv:Envelope>
  
```

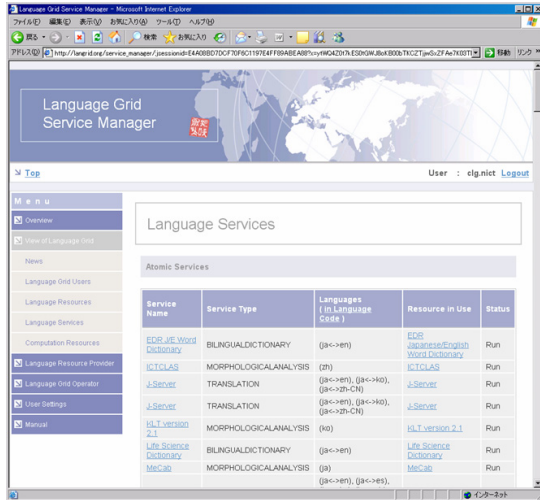
Annotations in Figure 5:

- Service ID: points to CLWT
- URI of the invoked Web service: points to http%3a%2f%2flanres%2enict%2ego%2ejp%2flangrid%2fservices%2fJServer,CLWT

Figure 5. Binding information in SOAP header

### 4. Language grid service manager

As shown in Figure 6, the service manager is a web-based tool for the Language Grid Users and the Operator to access various interfaces of the Language Grid. In Figure 6, the service manager presents a list of language services registered in the Language Grid when requested through the interface. The list consists of service name, service type, supported language pairs, constituent services, and status of the services. The right side of Figure 6 shows three composite services; two-hop translation consisting of J-Server and WEB-Transter, backtranslation using WEB-Transter, and backtranslation using J-Server. Currently, though the language services are listed up in the order of registration, they will be classified into four layers as proposed in Section 3.2 when the number of language services becomes significant. In this way, this tool allows easy management of language services, as well as computation resources, language resources. In this section, we focus on the management of language services.



(a) GUI of language grid service manager

Service Name	Service Type	Languages ( in Language Code )	Services in Use	Status
<a href="#">Multilingual Translation Service via J-Server at NICT</a>	TRANSLATION	(ja->en), (ja->zh), (ja->ko), (ja->fr), (ja->de), (ja->it), (ja->pt), (ja->es), (en->zh), (en->ko), (en->fr), (en->de), (en->it), (en->pt), (en->es), (zh->ko), (zh->fr), (zh->de), (zh->it), (zh->pt), (zh->es), (ko->fr), (ko->de), (ko->it), (ko->pt), (ko->es), (fr->de), (fr->it), (fr->pt), (fr->es), (de->it), (de->pt), (de->es), (it->pt), (it->es), (pt->es)	<a href="#">J-Server</a> , <a href="#">WEB-Transer</a>	Run
<a href="#">Backtranslation Using WEB-Transer</a>	BACKTRANSLATION	(ja-en->ja), (en-ja->en), (ja-zh->ja), (zh-ja->zh), (ja-ko->ja), (ko-ja->ko), (en-fr->en), (fr-en->fr), (en-de->en), (en-it->en), (it-en->it), (en-pt->en), (pt-en->pt), (en-es->en), (es-en->es)	<a href="#">WEB-Transer</a>	Run
<a href="#">Backtranslation Using J-Server</a>	BACKTRANSLATION	(ja-en->ja), (en-ja->en), (ja-zh->ja), (zh-ja->zh), (ja-ko->ja), (ko-ja->ko)	<a href="#">J-Server</a>	Run

(b) List of composite services

**Figure6. Language grid service manager**

#### 4.1. Management of atomic services

Language Grid Users, including Language Resource Providers, Computation Resource Providers, and Language Service Users, can register, update, and delete atomic services via the Language Grid service manager. In registering atomic services, they need to prepare service profiles, consisting of service ID, service name, service type, and supported languages, and WSDL files of the atomic services. When changing endpoints of the atomic services, the users need new WSDL files describing new endpoint addresses. An atomic service can be deleted only by the party that registered it.

The users also can associate other endpoints with the registered atomic services in order to distribute the loads across multiple endpoints. The added services are listed up as alternative endpoints to access in service invocation.

Moreover, the users can confirm what language resources the registered atomic services correspond to. By using this function, the users can invoke atomic services considering the access limits set against the corresponding language resources.

#### 4.2. Management of composite services

Language Grid Users can register, update, and delete composite services as well as atomic services via the service manager. To register composite services, they need to prepare abstract workflow files in addition to information necessary to register atomic services. If the same abstract workflow file has already been

deployed on the Language Grid, the users do not have to register a new abstract workflow file. They have only to invoke the composite service corresponding to the abstract workflow on the Language Grid with binding information.

Users often have trouble in finding the service type of the constituents of composite services because they do not access the constituent services directly. To solve this problem, the service manager provides not only a service profile but also a list of constituent services.

### 5. Conclusion

This paper introduces a four-layered language service architecture into the Language Grid: user adaptation, coordination, combination, and system adaptation layers. The essence of the architecture is as follows.

- User adaptation layer can adapt translation services to users by combining them with community-oriented language resources, such as multilingual community dictionaries, multilingual parallel texts, and community-oriented paraphrase dictionaries; the goal is to increase translation quality of community-specific sentences and the readability of the translation results.
- Coordination layer can coordinate multiple translation services so that they can share the context within a sentence, text or conversation.
- Combination layer can combine multiple translation services to make translation possible among any two languages.

- System adaptation layer can simplify or adjust input sentences and change abbreviations to standard nomenclatures so that translation services can translate them easily.

The architecture enables the users of the Language Grid to create various language services by choosing workflows in each layer and binding language resources to the workflows. As a result, users can easily combine multiple language resources without describing any complex workflows in BPEL4WS. However, as the number of language services is projected to keep increasing, the proposed four-layered language service architecture will not be enough to manage language services in the future. As the next step, we will develop a language service repository that increases the reusability of the accumulated language services.

## References

- [1] N. Calzolari, A. Zampolli, and A. Lenci, "Towards a Standard for a Multilingual Lexical Entry: The EAGLES/ISLE Initiative", *CICLing*, 2002, pp. 264-279.
- [2] K. Choukri, "European Language Resources Association History and Recent Developments", *SCALLA Working Conference*, 2004, KC 14/20.
- [3] C. Fellbaum and P. Vossen, "Connecting the Universal to the Specific: Towards the Global Grid", *Intercultural Collaboration*, Lecture Notes in Computer Science 4568, 2007, pp. 1-16.
- [4] Y. Hayashi, T. Declerck, P. Buitelaar, and M. Monachini, "Ontologies for a Global Language Infrastructure", *Proceedings of the First International Conference on Global Interoperability for Language Resources*, 2008, pp. 105-112.
- [5] Y. Hayashi and T. Ishida, "A Dictionary Model for Unifying Machine Readable Dictionaries and Computational Concept Lexicons", *International Conference on Language Resources and Evaluation*, 2006, pp.1-6.
- [6] T. Ishida, "Language Grid: An Infrastructure for Intercultural Collaboration", *IEEE/IPSJ Symposium on Applications and the Internet*, 2006, pp. 96-100.
- [7] T. Ishida, S.R. Fussell, and P. Vossen Eds., *Intercultural Collaboration*. Lecture Notes in Computer Science, 4568, Springer-Verlag, 2007.
- [8] T. Ishida, A. Nadamoto, Y. Murakami, R. Inaba, T. Shigenobu, S. Matsubara, H. Hattori, Y. Kubota, T. Nakaguchi, and E. Tsunokawa, "A Non-Profit Operation Model for the Language Grid", *Proceedings of the First International Conference on Global Interoperability for Language Resources*, 2008, pp. 114-121.
- [9] R. Khalaf, N. Mukhi, and S. Weerawarana, "Service-Oriented Composition in BPEL4WS", *Proceedings of the World Wide Web Conference*, 2003.
- [10] Y. Murakami, T. Ishida, and T. Nakaguchi, "Infrastructure for Language Service Composition", *Proceedings of the Second International Conference on Semantics, Knowledge, Grid (SKG-06)*, 2006.
- [11] S. Sakai, M. Gotou, M. Tanaka, R. Inaba, Y. Murakami, T. Yoshino, Y. Hayashi, Y. Kitamura, Y. Mori, T. Takasaki, Y. Naya, A. Shigeno, S. Matsubara, and T. Ishida, "Language Grid Association: Action Research on Supporting the Multicultural Society", *Proceedings of International Conference on Informatics Education and Research for Knowledge-Circulating Society (ICKS-08)*, 2008.
- [12] P. Vossen, "Introduction to EuroWordNet", *Computers and the Humanities*, 1998, Vol. 32, No. 2-3, pp. 73-89.
- [13] N. Yamashita and T. Ishida, "Effects of Machine Translation on Collaborative Work", *International Conference on Computer Supported Cooperative Work*, 2006, pp. 515-523