

Language Grid Toolbox: Open Source Multi-language Community Site

Masahiro Tanaka, Yohei Murakami, Donghui Lin
National Institute of Information and Communications
Technology (NICT)
3-5 Hikaridai, Seika-Cho, Soraku-Gun,
Kyoto-Fu, 619-0285, Japan
{mtnk, yohei, lindh}@nict.go.jp

Toru Ishida
Department of Social Informatics,
Kyoto University,
Yoshida-Honmachi, Sakyo-Ku
Kyoto-Shi, 606-8005, Japan
ishida@i.kyoto-u.ac.jp

Abstract— With the development of the Internet environments, more and more machine translation tools have become available on the Web, which provides the opportunity for some multilingual communities to use machine translation for communication. However, there exist several major problems in machine translation tools available on the Internet. 1) Translation for specific terms or sentences within communities is always of low quality. 2) Most of the available machine translation tools only provide basic functions of translating sentences, documents or Web pages, and lack necessary communication functions for community users. 3) It is difficult to conduct collaborative work among community users. 4) There is no support for creating customized multilingual environments based on unique community requirements. To address the above issues, we develop the Language Grid Toolbox which enables easy creation of multilingual community sites and customized multilingual environments for communities. For example, Toolbox has the basic function of creating language resources like dictionaries, which are used to combine with atomic translation services to improve the translation quality. Further, since Toolbox is developed as open source software and provides many basic APIs that can be used for communication, customized functions for communities can be easily developed. Several customized communication tools that are extended from Toolbox basic modules have already been implemented by universities and local government.

Keywords- Multi-language community, the Language Grid, language service, customization

I. INTRODUCTION

In some communities like international communication promotion organizations where participants are from various nations with various different languages, English might not always be the common language to be used for communication. In some situations, machine translation is used as an alternative media for communication. With more and more machine translation tools (e.g., the Google translation service) becoming available on the Internet, people start to consider the effectiveness of using machines translation tools for communication [1] [3] [8].

However, there exist several major problems in using machine translation tools available on the Internet for communication within communities.

First, the translation quality for community specific terms or sentences with such terms is always very low. Some commercial machine translation products or machine translation service available on the Web enables users to choose specific domains before translation. However, users are always not allowed to register their own specific community terms and combine with the machine translation service to get appropriate translation results.

Next, most current machine translation tools only provide basic functions of translating sentences, documents or Web pages while lack necessary communication functions for community users. To enable communication with people with different mother tongue through machine translation service, communication functions are required.

Further, it is always difficult to conduct collaborative work among community users. In multilingual communities, although multiple community users always conduct translation tasks together, it is difficult to process the translation tasks by sharing task information and the progress status among users.

Finally, current machine translation tools always lack support for creating customized multilingual environments based on different requirements from communities.

To address the above issues, we develop the Language Grid Toolbox (also referred to as Toolbox) which provides basic functions like language resource creation, text translation, multilingual BBS and Web creation that are required for multilingual communities. All the functions are realized by using language services on the Language Grid, such as machine translation services, dictionary services, morphological analysis services and so on.

By using the basic function of creating language resources like dictionaries in Toolbox, community specific terms can be registered in dictionaries, which can be used to combine with atomic translation services to improve the translation quality. In multilingual BBS, messages can be exchanged among community users and translated into other languages. Web creation function enables users to register translation result of Web contents as templates and share those templates among community users.

Toolbox is developed as open source software and provides APIs for basic functions. By this means, customized functions for communities can be easily developed. Several customized functions have already been implemented by universities and local government and embedded into Toolbox.

The rest of this paper is organized as follows. We first introduce the Language Grid which is the basis of Toolbox. Next, we describe the design concept, the architecture, basic functions, APIs of Toolbox. Further, we introduce several customized tools for communities developed based on Toolbox API, followed by the conclusion.

II. THE LANGUAGE GRID

To support users to create customized language services for their activities, the Language Grid¹, which is an infrastructure that is built on the top of the Internet, has been developed. The Language Grid provides an environment where users can share language resources developed by both professionals and end users in various application fields. Users can register wrap language resources as Web services to register them in the Language Grid [2][3].

The Language Grid takes the collective intelligence approach, and therefore the platform can grow only through the voluntary efforts of users [4]. The more users provide resources, the more fully they can utilize the benefits of the re-sources. Thus the platform should allow users to create services and share them via the Language Grid. Conceptually, the Language Grid has two main structures: horizontal and vertical. The horizontal grid concerns the combination of existing bilingual dictionaries or machine translation systems for various languages. The vertical grid concerns specific scenarios of intercultural collaboration activities, which require specialized language services including jargon handling.

The Language Grid is built on the P2P Grid Infrastructure [6], which connects two kinds of servers (core nodes and service nodes). Core nodes manage all requests to language services, while service nodes actually invoke atomic services. If the requested service is a composite one, core nodes invoke a corresponding Web service workflow that includes one or more atomic services. All the language resources and language services can be managed by using the Language Grid Service Manager².

Different types of collaboration tools have been developed using the language services provided by the Language Grid. Language Grid Playground³ provides easy access to the Language Grid to try a variety of registered language services through a Web browser.

In this paper, we aim to provide a customizable framework for collection of modules to support multilingual communication in a community so that users can use it to start their own services, such as multilingual BBS and multilingual dictionary creation. To provide such a customizable framework for intercultural collaboration, we develop the Language Grid

¹ <http://langrid.nict.go.jp>

² http://langrid.org/service_manager/

³ <http://langrid.org/playground/>

Toolbox as open source software, which can be easily extended by developing modules by user communities to meet different requirements for intercultural collaboration. In the rest of the paper, we will introduce architecture, basic functions, customization features and applications of Language Grid Toolbox.

III. DESIGN CONCEPT

Toolbox is a Web application which provides support tools for multilingual communication. Toolbox was developed by extending XOOPS, an open source CMS (Contents Management System). Figure 1 shows a screenshot of Toolbox.

Toolbox is a Web application which provides support tools for multilingual communication. Toolbox was developed by extending XOOPS, an open source CMS (Contents Management System). Figure 1 shows a screenshot of Toolbox.

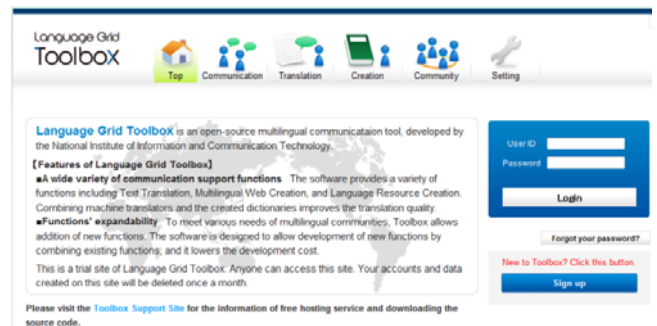


Figure 1. A Trial Site of Language Grid Toolbox

A. System Goal

The goal of Toolbox is to provide customizability to support a wide variety of collaboration in multilingual communities. Customization for multilingual communities is classified into “Customization of language resources” for improving translation quality and “Customization of functions” for facilitating collaboration. Toolbox focuses on supporting collaboration in a multilingual community. In this regard, the goal of Toolbox is different from that of Language Grid Playground [7], which focuses on showing various combinations of language services on the Language Grid.

The former is achieved by a function which allows community members to accumulate translations of particular terms and expressions of the community. The translation quality generally declines if the input sentences contain particular terms and expressions of a community because a machine translator for general purposes cannot correctly translate such terms and expressions. Therefore we improve translation quality by combining the translator with dictionaries which contain terms and expressions accumulated in a community.

On the other hand, the latter is realized by adding new functions or integrates existing functions. Functions of existing multilingual communication tools are often too general to meet needs in the field and have made collaboration less effective. Otherwise, they are too specialized in a certain field and lacks versatility.

Therefore we first identified four basic modules for multilingual communities: language resource creation, translation, communication, and collaboration. Then we provided APIs of the modules in order to reduce development cost of collaboration support functions specialized in each field.

B. System Architecture

Figure 2 shows the system architecture of Toolbox.

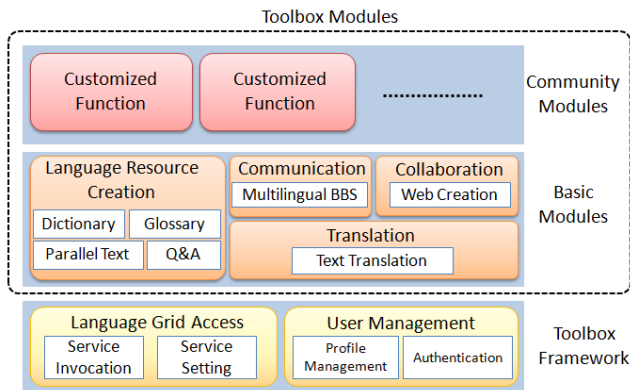


Figure 2. Architecture of Language Grid Toolbox

As shown in Fig. 2, Toolbox consists of two main types of components: Toolbox modules, which provides functions for supporting multilingual communication, and the Toolbox framework, which provides underlying functions

The Toolbox framework works as an infrastructure of all Toolbox modules. It comprises Language Grid Access function and User Management function.

Language Grid Access function allows Toolbox modules to invoke language services on the Language Grid. It manages service binding information, which defines a combination of atomic services invoked from a composite service, and invokes the composite service using the service binding information. Using the APIs of Language Grid Access, even a developer who does not know Web service technologies and the Language Grid can invoke services on the Language Grid.

Language Grid Access function also provides a GUI which allows end users to configure translation settings for each Toolbox module (Figure 3). A user can save his/her own settings. The developer of a Toolbox module can invoke the translation service according to a given language pair and the user's setting only by referring the settings.

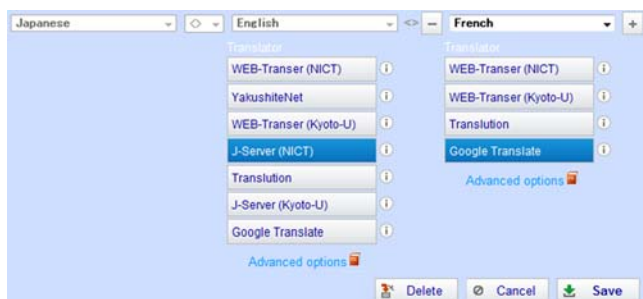


Figure 3. Services Setting

User Management function allows Toolbox modules to manage members of a multilingual community. It also handles authentication of Toolbox users, and shows the list of members of a community to let them know the state of the community and information of other members. User profiles are managed by this function and can be defined for each community. Community members can search for other members based on the profiles.

There are two types of Toolbox modules, basic modules that implement the basic functions for multilingual collaboration, and community modules that provide special functions based on the community requirements.

Basic modules can be categorized as four types: language resource creation that enables community users to create language resources for the purpose of improving translation quality, translation that enables users to translate text information multilingually by using created language resources and language services on the Language Grid, communication that enables users to communicate with each other, collaboration that enables multiple users to conduct collaborative work by sharing contents (including the task result from users). Basic modules provide API for using the basic module functions from other modules.

Community modules use functions of basic modules through APIs, and implement community specific functions. In community modules, implementation cost can be significantly reduced since the implementation mainly focuses the part of community specific functions. Moreover, usage of APIs from multiple basic modules enables easy implementation of community modules that have composite functions. Further, community modules also provide APIs so that new modules can be built even above them.

C. Module Customization

Customization of community modules follows the steps below based on the internal structure of the module with the dotted line around.

1. Design CHI from use scenario
2. Create layout and design of screen
3. Get data using existing API
4. Implement customized processes
5. Package as Toolbox module

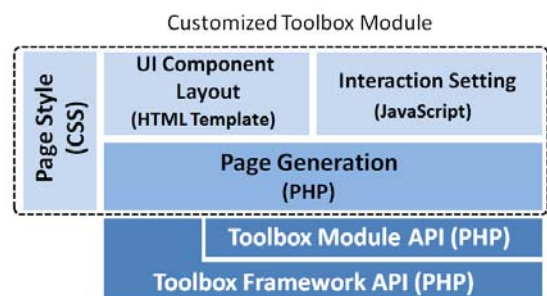


Figure 4. Structure of Toolbox module

Customization begins with designing interaction between users and Toolbox or among users via Toolbox based on use cases created by interviewing users in the field. Next interaction settings are implemented in JavaScript by using a mock-up which represents the designed interaction. Static layout of components and page design are implemented as HTML templates, which have parameters to be instantiated by PHP programs and suit to displaying an array structure.

Then the developer needs to check if values for a parameter in HTML templates can be obtained by existing APIs or not. If existing APIs can obtain the required values, the developer installs the module which provides the APIs. If not, he/she needs to implement customized processes. The screen which is shown to users are generated by Page Generation component getting values for parameters in HTML templates. Finally all the required components are packaged as a Toolbox module.

IV. BASIC MODULES

Since Language Grid Toolbox is designed for the purpose of supporting community users to create their own multilingual community sites, it provides the basic functions based on the most important requirements in multilingual communities including the language resource creation, translation, communication and collaboration functions.

A. Language Resource Creation

Quality of machine translation services can always be improved with customization in specific domains of multilingual communities. Therefore, creating and sharing language resources is very important for communities to conduct multilingual communication and collaboration. To provide easy customization environments for communities to use the Language Grid, Language Grid Toolbox provides the language resource creation as basic function that enables community users to create and share dictionaries, parallel texts, Q&As and glossaries. Figure 5 shows the screenshot of the language resource creation (dictionary).

Moreover, features like deployment of language resources as language services are important for community users. For example, deployed community dictionary services can be shared and used as local dictionary services on Language Grid Toolbox, which can be combined with machine translation services as well as the global dictionary services provided by the Language Grid. By this means, users can use more and more alternative language services created by the communities.



Figure 5. Language Resource Creation (Dictionary)

B. Communication

Multilingual communication is another important requirement when creating multilingual community sites. Language Grid Toolbox provides a basic communication called multilingual BBS that enables community users to communicate with each other with their mother tongues since the contents of BBS are translated multilingually by language services on the Language Grid. Like the basic translation function above, community users can also customize the usage of translation services. Moreover, an important feature of the multilingual BBS is that users can modify the machine translation results and improve the quality of multilingual contents. Figure 6 shows the screenshot of the multilingual BBS.



Figure 6. Multilingual BBS

C. Collaboration

Language Grid Toolbox also provides Web creation as a basic function for community users to collaborate to create multilingual Web pages together. The Web Creation function provides translation templates to support creation of multilingual Web contents. When a user translates Web contents, the user can apply a translation template, and the part corresponding to the template will be translated as specified by the user. Community users can create HTML templates together and share them to multilingualize the Web contents. The translation services used for Web creation can also be customized by users like the text translation and multilingual BBS functions. Figure 7 shows the screenshot of the Web creation function.

Community users can customize above basic functions, or combine the internal basic functions and external functions for their own requirements in multilingual activities. For example, users can extend the basic text translation function to collaborative translation function, the multilingual BBS function to multilingual discussion function, and so on. We will introduce the customization of Language Grid Toolbox in the following sections.



Figure 7. Web Creation



Figure 8. Text Translation

V. TOOLBOX APIS

The core components of Toolbox modules are implemented in PHP. This is the reason Toolbox APIs are provided as methods of PHP classes. Each basic module has a corresponding class which provides APIs as methods of the class APIs. Customized Toolbox modules can use APIs by creating an object for the class.

In this section, we introduce APIs of each basic modules and the Toolbox framework.

Language Resource Management API: This API set provides access to language resources, such as multilingual dictionaries, parallel texts, Q&As, and glossaries. Some APIs are common in any types of resources, and some are specific for each type. APIs which are common in any types include getting information of language resources, creating/deleting a language resource, and modifying properties of language resources such as supported languages. APIs specific for each type include adding/deleting/updating/searching for records. An API which deploys/undeploys a language resource as a Web service is also provided.

Multilingual BBS API: This API set allows to access to the multilingual BBS in Toolbox. The same operations as a user can perform via GUI are provided by APIs. For example, other modules can get the contents on the BBS, post new contents, and manages category hierarchies of messages using the APIs. Using the APIs, developers can easily construct a new module which contains functionality of the Multilingual BBS. The translation of posted message is performed by the API of Text Translation, not by this API set.

Text Translation API: This API set provides translation and back translation by machine translation services on the Language Grid. The APIs refer to translation settings stored in Language Grid Access module. Extended translation functions, such as translation with caching, are also implemented as an API of Text Translation module. This API set underlies other modules which use translation as the interface of Language Grid Access function in the Toolbox framework.

Functions of the Toolbox framework are also provided as APIs which can be called by Toolbox modules. We show the APIs of Language Grid Access function and User Management function.

Language Grid Access API: This API set allows Toolbox modules to access language services on the Language Grid. Using the APIs, Toolbox modules can invoke translation service combining machine translation services, dictionary services, and morphological analysis services. Translation settings are defined for each tool in Toolbox, such as BBS and Text Translation, and consist of a set of language paths for translation and services assigned to the paths. Given the source language and the target language, the translation API invokes the services assigned to the language pair. Although the standard method of invoking language services on the Language Grid requires a client to describe complex service settings in SOAP header, this API set provides much simpler way to access the Language Grid. Moreover, it can refer translation settings created by end users via GUI.

User Management API: This API set allows Toolbox modules to get a list of users registered to Toolbox and get/set the users' profiles. Such functions are required for community members to share information of each other and to know the status of other members.

Table 1 shows some of APIs provided by basic modules and the Toolbox framework.

Table 1 Overview of Toolbox APIs

Functions	API
Language Resources Management	Create/Edit/Remove a language resource, get a list of language resources, add/remove/get/update records in a language resource, deploy a language resource
Multilingual BBS	Post/Reply/Get/Edit/Remove message, create/get/edit/remove category hierarchies of messages, search for messages, revise the result of translation, manage history of revisions of translation result

Text Translation	Translate/Back-translate a given string.
Language Grid Access	Edit translation settings defined for tools in Toolbox, invoke language services on the Language Grid, get a list of language services on the Language Grid
User Management	Get a list of users, get the current user, set/get user profiles

We show the definition of translation API provided by the Language Grid Access module as an example of APIs described above.

Table 2 API Example

Invoke translation service	
TranslationResult[] translate(String \$sourceLang, String \$targetLang, String \$source, String \$translationBindingSetName)	
Description	This API translates the given string following the translation setting identified by the fourth arguments. The translation settings specify translation services, dictionary services, and morphological analyzer services for each translation path. The services which are invoked by this API are determined based on the source language and the target language.
Arguments	\$sourceLang: Source language \$targetLang: Target language \$source: String to be translated \$translationBindingSetName: Reference to translation settings
Return value	The return value contains the translation results and invocation information such as the list of services actually invoked and response time of invocation. If multiple translation settings are defined for the given pair of the source language and target language, the same number of translation results are returned.

VI. MODULES CUSTOMIZED FOR COMMUNITY

Based on the basic modules of Language Grid Toolbox and the provided API, universities and local government at Kyoto city of Japan are putting efforts to develop customized modules to meet requirements of communities. In this section, we introduce functions, applications and use cases of those customized modules.

A. Discussion Module

The discussion function is developed by extending the multilingual BBS to enable community users to communicate with each other while sharing contents like images, maps and so on.

By using the discussion function, a user can share contents like images and maps, and link the shared contents with the

messages posted on the multilingual BBS. By this means, the posted messages and shared contents can be displayed together on the screen. Moreover, markers can be set on the shared images or maps so that users can discuss about the marked part of the contents more smoothly.

The discussion function is developed by Kyoto University, Japan and is used to support foreign students who cannot speak Japanese in the community of graduate school at Kyoto University. When foreign students participate in the research seminars in a laboratory, Japanese students who will present in Japanese always upload their presentation slides as the format of images in the discussion function to support the foreign students. Since the discussion function is extended from the multilingual BBS, foreign students can participate in the discussion in the BBS using their mother tongues, and Japanese students always summarize what the presenter talks about. Therefore, multilingual communication can be realized between Japanese students and foreign students.

In the discussion function, API of multilingual BBS is used to get the messages and handle the post. For other parts of the function like display of hierarchical categories of messages, translation through Language Grid and multilingual display are the same as those in the multilingual BBS.

B. Collaborative Translation Module

The collaborative translation function is developed by extending the Language Grid Toolbox basic function of text translation to enable collaborative work for translation tasks by multiple community users.

Progress of translation tasks can be recorded in the collaborative translation function. The status of each unit in the document for translation can be set as “not modified”, “in progress” or “finished”. The status of “not modified” means that the translation result of the unit is from machine translation service and has not been modified yet. The status of “in progress” means that the translation result of the unit is from machine translation service and is currently being modified by the community users. The status of “finished” means that the translation result has been modified or confirmed by the community users. By sharing the progress status of the translation tasks among community users, the whole tasks can be appropriately assigned to multiple users for collaborative work.

The collaborative translation function is developed by Kyoto city and is expected to be used to support translation pamphlet in shops and temples in Kyoto. Since bilingual translators are not always available in such areas, and the cost of translations of highly-trained bilingual individuals are always high. To reduce the translation cost, translation process that combines machine translation service on the Language Grid and human tasks has been proposed [5]. In the proposed process, machine translation service is first to be used to get the initial machine translation result of the source document, then monolingual people is introduced to revise the machine translation result to make the result more fluent, and at last bilingual translator is in charge of confirm and re-translate part of the intermediate result by the monolingual people. Since the translation process is progressed by collaboration of users of

different roles, the collaborative translation function can be applied because it can record the progress status of the translation tasks.

Moreover, the collaborative translation user interface provides the functions of easily searching and registering dictionaries and parallel texts without stopping the translation tasks.

In the collaborative translation function, API of text translation is used to execute the translation. Searching and registration of dictionaries and parallel texts are developed based on the API of the language resource management.

C. Q&A Web Interface Module

The Q&A Web interface function is developed by combining the multilingual Q&A creation function and the multilingual BBS function. By using the Q&A Web interface function, Toolbox community users can collaborate to create Q&A contents for users outside the community.

The Q&A Web interface site enables users that do not belong the Toolbox community to search Q&As in Toolbox and post new questions. When a new question is posted, it will be automatically registered as Q&A in Toolbox, and added as a new topic in the multilingual BBS. Answers will be created based on the discussion among Toolbox users. Created answers can be searched by the Q&A Web interface if they are added into Q&As in Toolbox.

The Q&A Web interface function is also developed by Kyoto University. Although Q&As and other information for foreign students are accumulated in the Toolbox site that is operated by Kyoto University, it is not open to users who are not community users in the graduate school. The Q&A Web interface function enables searching Q&As and asking questions from all users including students currently in foreign countries who want to come to Japan for study

The Q&A Web interface function is implemented by executing APIs of multilingual BBS and Q&A modules of Toolbox from the site that provides user to search Q&As and register questions.

D. Reception Module

The reception function is developed to support communication using multilingual Q&A registered in Toolbox, which is always used to support customers in the front desk at shops. Customers can select the Q&A category and questions to find answers. Moreover, the reception function is composed with the glossary registered in Toolbox. When the words registered in the glossary are included in answers, the meaning of the words can be displayed by hovering the mouse cursor.

The reception function is developed by Kyoto city and is expected to be used to support foreign customers to communicate with Japanese staffs at the front desk at shops in Kyoto.

The reception function is implemented by using APIs of multilingual Q&A and multilingual glossary.

VII. CONCLUSION

In this paper, we proposed Language Grid Toolbox, which is multi-language communication tool for community. Language Grid Toolbox provides basic functions for multi-language communication. Moreover, it is provided as an open source software to allow communities to extend the basic functions based on suit their needs.

The contributions of this work are as follows:

- Toolbox allows community members to improve translation quality by creating language resources for their own community and sharing translation results.
- APIs of Toolbox modules make it easier for developers to build a new function which suits needs of each community.

Various organizations have already registered their own Multilingual dictionaries and Q&As to their Toolbox. Moreover, a municipality and a university have developed new tools which are specialized for their needs.

As the next step, we will build an open source community of Toolbox where various organizations join and develop new modules.

REFERENCES

This work was partially supported by Strategic Information and Communications R&D Promotion Programme from Ministry of Internal Affairs and Communications.

REFERENCES

- [1] Inaba, R., Murakami, Y., Nadamoto, A., and Ishida, T. Multilingual Communication Support Using the Language Grid. *International Workshop on Intercultural Collaboration (IWIC-07)*, pp. 118-132, 2007.
- [2] Ishida, T. Language Grid: An infrastructure for intercultural collaboration. *IEEE/IPSJ Symposium on Applications and the Internet (SAINT-06)*, pp. 96-100, keynote address, 2006.
- [3] Ishida, T. Intercultural Collaboration Using Machine Translation. *IEEE Internet Computing*, pp. 26-28, 2010.
- [4] Ishida, T., Murakami, Y., Inaba, R., Shigenobu, T., Lin, D., and Tanaka, M. The Language Grid: Creating Customized Multilingual Environments. *International Conference on Global Interoperability for Language Resources (ICGL-10)*, 2010.
- [5] Lin, D., Murakami, Y., Ishida, T., Murakami, Y., and Tanaka, M. Lessons Learned from Composing Web Services and Human Activities. *7th International Joint Conference on Service Oriented Computing (ICSOC-09)*, 2009.
- [6] Murakami, Y., and Ishida, T. A Layered Language Service Architecture for Intercultural Collaboration. *International Conference on Creating, Connecting and Collaborating through Computing (CS-08)*, 2008.
- [7] Sakai, S., Gotou, M., Morimoto, S., Morita, D., Tanaka, M., Ishida, T., and Murakami, Y. Language Grid Playground: Light Weight Building Blocks for Intercultural Collaboration. *International Workshop on Intercultural Collaboration (IWIC-09)*, pp. 297-300, 2009.
- [8] Yamashita, N., and Ishida, T. Effects of Machine Translation on Collaborative Work. *International Conference on Computer Supported Cooperative Work (CSCW-06)*, pp. 515-523, 2006.