

Master Thesis

**Supporting Multilingual Collaboration
for Wikipedia Translations**

Supervisor Professor Toru ISHIDA

Department of Social Informatics
Graduate School of Informatics
Kyoto University

Masanobu ISHIMATSU

February 9, 2011

Supporting Multilingual Collaboration for Wikipedia Translations

Masanobu ISHIMATSU

Abstract

In Wikipedia, the largest encyclopedia on the Internet, a huge amount of knowledge is shared among users. However, differences in the number of articles among different language versions of Wikipedia represent an important issue. In order to solve the current imbalance of knowledge present in different languages, some users translate existing articles from one language to create new articles in another. These are known as "Wikipedia Translations."

Unfortunately, there are numerous instances of highly culture-specific articles in Wikipedia, which cannot be translated by individual translators. In such cases, collaborations among translators and experts are required. The author refers to this process as "Collaborative Translations." For those who are not bilingual, it is difficult to explain culture-specific knowledge in languages other than their native tongue. The need for a multilingual environment arises which enables such users to communicate and collaborate with translators in their native language. In order to construct this kind of an environment, the author solves two problems:

Designing a multilingual communication system for collaborative translations: The author will design a multilingual BBS system based on machine translations. Because of the features of Wikipedia Translations, the author has to consider two problems: First, the difficulty in using pre-defined languages for machine translation, since Wikipedia users with unspecified language abilities can collaborate on Wikipedia translations. Second, there can be many mistranslations of specific words in highly culture-specific articles that occur as a result of machine translators. Furthermore, the same specific words can be translated different in different articles.

Controlling multilingual collaborations based on collaborative translation protocol: Wiki users are loosely coordinated volunteers, and they have various language skills. In the Wikipedia translation experiment, there were also several loosely coordinated participants. The author observed that users

could not view progress of translation tasks and discussions in their collaborative translations because many discussions occurred in parallel and they could not follow all discussions.

To resolve the first problem, the author designed a multilingual collaboration system on the Wikipedia system. In order to design this system, the author added multilingual abilities to LiquidThreads, a BBS running on the Wikipedia system. In order to solve the problem of difficulties arising from the use of predefined languages for machine translation, the author designed a system which translates messages in the viewing phase rather than in the posting phase. This system can answer viewing requests in any language. Moreover, the author designed the system allowing translations of sentences by machine translators combined with dictionaries, which influence one page at a time.

To resolve the second problem, the author analyzed logs of the Wikipedia translation experiment and created a protocol for multilingual collaborations in Wikipedia translations. The author divided users into two groups, the target language side users and the source language side users. The author then designed the protocol for collaborations between these two groups. Furthermore, the author implemented a control system of multilingual collaborations based on this protocol and integrated this system with the multilingual BBS.

Contributions of this research are as follows:

Designing multilingual BBS and a dictionary for each page and combining them with machine translation: The author designed the Multilingual LiquidThreads, a multilingual BBS which translates messages in the viewing phase. The author also designed PageDict, a dictionary system for use with machine translations. These systems are opened as MediaWiki extensions.

Designing the collaborative translation protocol and developing the collaboration control system based on the utterance intentions: The author designed the collaborative translation protocol by analyzing logs of the collaborative translation experiment. This protocol is represented by a state transition diagram in which each state will be transited by utterance intentions. The author developed the collaboration control system based on the protocol and integrated it along with the Multilingual LiquidThreads.

Wikipedia 翻訳のための多言語協働支援

石松 昌展

内容梗概

インターネット上の百科事典である Wikipedia には、世界各国のユーザの知識が蓄えられ、その量はますます増えてきている。しかしながら、Wikipedia の発達に伴って、各言語ごとに記事数が大きく異なっていることが問題視されるようになってきた。そこで、言語による知識の偏在を解決するために、記事の翻訳が行われている。これを Wikipedia 翻訳と呼ぶ。

しかし、Wikipedia の記事は専門性や文化依存度の高いものが多く、その翻訳には、記事に関する知識を持つ者と翻訳の知識を持つ者が協力する必要がある。これを協働翻訳と呼ぶ。ただし、バイリンガルではない原文側のユーザには文化依存性のある文を翻訳先言語で説明することは困難である。このため、文化依存性のある文の翻訳には、ユーザが母国語を用いながらコラボレーションできる環境が必要である。この環境の構築実現において取り組むべき課題は以下の2つである。

翻訳のための多言語コミュニケーション環境 多言語コミュニケーションを実現するために、機械翻訳を用いた多言語掲示板システムを構築するが、Wikipedia は不特定多数の言語圏のユーザが利用するという特徴を持つため、あらかじめ言語を想定した翻訳システムを構築することが難しいという問題がある。また、機械翻訳による多言語コミュニケーション環境では、一般的に専門用語の誤訳や、非一貫性によりユーザのコミュニケーションに混乱が生じることが知られている。専門性や文化依存度が高い Wikipedia の記事の翻訳では機械翻訳の対応できない用語が多数出現し、また同じ用語であっても記事ごとに訳語が異なるという問題がある。

協働翻訳プロトコルによる多言語コラボレーションの制御 Wikipedia の記事の翻訳には不特定多数のユーザが携わることが想定される。我々の行った協働翻訳実験では、原文側のユーザと翻訳側のユーザが複数人ずつ存在したが、彼等のコミュニケーションにおいて翻訳のためのコラボレーションにおけるプロトコルが明確に定められていないために、協働翻訳作業が中断されてしまうという問題があった。また、機械翻訳の誤訳による発話意図の誤解によってコラボレーションが中断されるという問題もあった。

上記2つの問題を解決するために、以下の方策を取る。

まず、一つ目の問題を解決するために、Wikipediaのシステム上で動作する多言語コラボレーション環境を構築する。構築に当たっては、Wikipediaのシステム上で動作する掲示板システムであるLiquidThreadを多言語化する。Wikipediaは、不特定多数の言語圏のユーザが利用するという特徴を持つため、あらかじめ言語を想定した翻訳システムを構築することが難しいという問題に対しては、投稿時ではなく閲覧時にユーザの要求する言語に応じて翻訳を行うという方策を取る。こうすることで、ユーザの要求する任意の言語に対応することができる。また、Wikipediaの各記事における専門用語の一貫した翻訳を実現するために、記事ごとに辞書を作成し、その記事に関連するページにおける翻訳のみに対応する辞書連携翻訳を行う仕組みを確立する。

次に、二つ目の問題を解決するために、Wikipedia記事の協働翻訳実験で得られた対話ログを分析し、多言語コラボレーションによる協働翻訳プロトコルを明らかにする。分析ではユーザを原文側と翻訳側という二つの役割によって分類し、翻訳文完成にいたるまでの両者の多言語コミュニケーションにおけるプロトコルを明らかにする。さらに、ここで得られた協働翻訳プロトコルを用いて多言語コラボレーションを制御する機能を構築し、先に開発した多言語掲示板システムに統合する。

本研究の貢献は以下の2つである。

多言語コミュニケーション環境と記事ごとの専門用語辞書作成、辞書連携翻訳機能の実現 閲覧要求時に翻訳を行う多言語掲示板システムである多言語LiquidThreadsを構築した。また、記事ごとに異なる専門用語に対応するため、その記事関連の翻訳のみに影響を及ぼす辞書連携翻訳システムであるPageDictを構築した。さらに、これらのシステムはMediaWiki上で動作するシステムとして一般公開している。

協働翻訳プロトコルの提案と発話意図タグに基づく状態管理機能の導入 協働翻訳実験の分析から原文側と翻訳側の多言語コラボレーションによる協働翻訳プロトコルを明らかにした。このプロトコルは発話意図によって状態を遷移するものであり、複数ユーザ間のコラボレーションを制御することができる。また、本プロトコルに基づいて、発話意図を明確に伝えるためのタグと状態管理を行う機能を多言語LiquidThreadsシステムに導入した。

Supporting Multilingual Collaboration for Wikipedia Translations

Contents

Chapter 1	Introduction	1
Chapter 2	Wikipedia Translations	5
2.1	Translations of Wikipedia Articles	5
2.2	Collaborative Translations by Monolingual Users	7
2.3	Wikipedia Translations Experiment	9
Chapter 3	Supporting Multilingual Communications on the MediaWiki	13
3.1	MediaWiki and LiquidThreads	13
3.2	Functional Requirements	15
3.3	Multilingual LiquidThreads	17
3.4	PageDict	26
Chapter 4	Collaborative Translation Protocol for Wikipedia Translations	29
4.1	Controlling Users' Actions based on the Protocol	29
4.2	Designing the Protocol	31
4.3	Collaborative Translation Protocol	40
4.4	Validation of the Protocol	41
Chapter 5	Supporting Multilingual Collaborations based on the Collaborative Translation Protocol	44
5.1	Functional Requirements	44
5.2	Implementation	45
Chapter 6	Internationally Expanding of the Multilingual LiquidThreads	50
Chapter 7	Conclusion	53
	Acknowledgments	54
	References	55

Chapter 1 Introduction

In Wikipedia¹⁾, the largest encyclopedia on the Internet, a huge amount of knowledge is shared among users. Wikipedia was established by using the Wiki system "MediaWiki," which enable all users to view and edit their contents through the Web. Although the Wikipedia project is one of the Wikimedia Foundation projects managed by the foundation, the contents are managed by core users called "Wikipedian" who manage the quality of articles and discuss the policy of Wikipedia. Wikipedians are volunteers who work by contributing time and effort to enrich the qualities of Wikipedia articles without any pay. There are also novices who contribute by reading articles out of interest, noting mistakes or omissions, and correcting them[1]. Currently there are about 300,000 editors who have edited English Wikipedia articles for more than 10 times. Approximately the same number of editors edit Wikipedia every month; with about 50,000 doing more than 5 edits and 5,000 doing more than 100 edits²⁾.

Collective knowledge in Wikipedia is the result of a huge number of edits by loosely coordinated volunteers who have various motivations. MediaWiki, the Wiki system, has functions which strongly support these users' activities. First of all, it has a user interface which supports easy editing of articles. For example, articles can be freely edited and stored by clicking the "save" button. Without learning computer languages, such as HTML, users can change the contents of a page. Second of all, it has revision and history functions. In the Wikipedia, all revisions of an article are saved and users can know who, when, and how other users changed the article by viewing the history page. Users can replace the newest version with the past revision if needed[2].

Wikipedia is a multilingual project. Several versions exist, including the English Wikipedia, French Wikipedia, Japanese Wikipedia, etc. Wikipedia, in each language, is managed by a different community, and its contents are also different. However, differences in the number of articles between different lan-

¹⁾ http://en.wikipedia.org/wiki/Main_Page

²⁾ http://en.wikipedia.org/wiki/English_Wikipedia

guage versions of the Wikipedia are perceived as problems by users, both by Wikipedians and the Wikimedia Foundation. For example, English Wikipedia has 3,500,000 articles, but Japanese Wikipedia has 730,000 articles while Chinese Wikipedia has only 340,000 articles¹⁾. Highly culture-specific articles are well written in the "native" language of the culture since editors are able to quite easily and correctly write details in their native language. There is also a rule of "Neutral Point of View"²⁾ in Wikipedia. For this rule, it is difficult for editors to write highly culture-specific articles in non-native languages. For these reasons, the Wikipedia project has not accomplished a system in which all knowledge is equally published in different languages.

To solve this imbalance of knowledge present in different languages, some Wikipedia users and outside community members support translations of articles from one language to another. These translations are called "Wikipedia Translations". Wikipedia has guidelines for translating and has pages for translation requests. Moreover, WikiBabel[3] and Google translator toolkits³⁾ have been developed and published as tools to support Wikipedia Translations.

However, the current approach of Wikipedia Translations is contents-based collective intelligence, in which bilinguals translate articles based on their own interests and knowledge. There are limitations to translating via this approach. And as a result, only articles in areas in which editors are interested in are translated, excluding many other areas which remain without translations.

In these cases, collaborations between translators and knowledge experts are required. The author calls this "Collaborative Translations". In this approach, translators translate articles through communications with knowledge experts who write the original articles and have the required culture-specific knowledge.

However, for non-bilingual users, it is difficult to explain the culture-specific knowledge in a language other than their native language. A multilingual environment which enables these users to communicate and collaborate with translators in their native language is needed. To construct this kind of an

¹⁾ http://meta.wikimedia.org/wiki/List_of_Wikipedias

²⁾ <http://en.wikipedia.org/wiki/NPOV>

³⁾ <http://translate.google.com/toolkit/>

environment, the author solves two problems as follows:

Designing a multilingual communication system for translations:

The author will design a multilingual BBS system based on machine translations for multilingual communications. As a result of Wikipedia Translations features, the author has to consider two problems: First, it is difficult to use pre-defined languages for machine translations because of Wikipedia users who have unspecified language abilities. Second, there can be many mistranslations by machine translators in translations of highly culture-specific articles which include a lot of specific words. There may be misunderstandings in multilingual communications for discussions about translations of Wikipedia articles. Furthermore, the same specific words can have different translations in different articles.

Controlling multilingual collaborations based on a collaborative translation protocol: Wiki users are loosely coordinated volunteers, and they have various language skills. In the Wikipedia translation experiment, there were also several loosely coordinated participants. The author observed that users could not view progress of translation tasks and discussions in their collaborative translations because many discussions occurred in parallel and they could not follow all discussions.

To resolve the first problem, the author designed a multilingual collaboration system based on the Wikipedia system. In order to design this system, the author added multilingual abilities to LiquidThreads, a bulletin board system running on the Wikipedia system. In order to solve the problem of difficulties arising from the use of pre-defined languages for machine translation, the author designed a system which translates messages in the viewing phase rather than in the posting phase. This system can answer viewing requests in any language. Moreover, the author designed the system allowing translations of sentences by machine translators combined with dictionaries, which influence one page at a time.

To resolve the second problem, the author analyzed logs of the Wikipedia translation experiment and created a protocol for multilingual collaborations in Wikipedia translations. The author divided users into two groups, the target

language side users and the source language side users. The author then designed the protocol for collaborations between these two groups. Furthermore, the author implemented a control system of multilingual collaborations based on this protocol and integrated this system with the multilingual BBS.

This paper is organized as follows: In Chapter 2, Wikipedia Translations and previous research relevant to this research are introduced. The Wikipedia Translation Experiment is also explained in more detail. In Chapter 3, the author introduces details of the designing stages of the system for multilingual communications in MediaWiki. In Chapter 4, the author introduces the collaborative translation protocol which was created through the analysis of logs of the experiment. In Chapter 5, details of the implementation are presented for controlling the collaboration system based on the protocol along with the explanation on how to integrate the system with the multilingual communication system. In Chapter 6, the author introduces the future open source project to establish multilingual environment in MediaWiki by developing and managing the Language Grid for Wikimedia, the Language Grid Extension and Multilingual LiquidThreads. Chapter 7 presents the conclusions.

Chapter 2 Wikipedia Translations

In this section, the author will introduce the current state of Wikipedia Translations, the importance of Wikipedia translations, research using translated articles in Wikipedia, and tools to support translators. Then, problems involving collaborations in Wikipedia Translations are introduced through analysis of the previous Wikipedia translation experiment.

2.1 Translations of Wikipedia Articles

As mentioned in Chapter 1, Wikipedia articles are often translated to create new articles in other language versions of the Wikipedia. This activity is called "Wikipedia Translations¹⁾". As of February 2011, one or two articles in the English Wikipedia are translated to several other languages per week²⁾. In the Wikipedia user pool, there are users who work as translators³⁾ and proof readers⁴⁾. In this way, Wikipedia Translations are already occurring as an activity of the Wikipedia.

When an article is translated into a new article in another language, the source article and the new article are connected by the "interlanguage link⁵⁾". Interlanguage links enable users to view articles on the same topic in various languages. There is a lot of research being conducted on using the symmetry of articles connected by interlanguage links[4, 5]. Moreover, there are activities to create multilingual dictionaries by extracting titles of articles which are connected by interlanguage links. In this way, Wikipedia Translation activities enable huge amounts of knowledge to be published in various languages and the creation of rich language resources.

There are several research methods for studying Wikipedia translations. The Wiki system has a feature that allows loosely coordinated users to collaborate and contribute based on their own motivations. Sharing multilingual knowl-

¹⁾ <http://en.wikipedia.org/wiki/Wikipedia:Translation>

²⁾ http://meta.wikimedia.org/wiki/Translation_of_the_week

³⁾ http://en.wikipedia.org/wiki/Category:Available_translators_in_Wikipedia

⁴⁾ http://en.wikipedia.org/wiki/Category:Available_proofreaders_in_Wikipedia

⁵⁾ http://meta.wikimedia.org/wiki/Interlanguage_link

edge under these conditions, one research points out that previous methods of traditional translations are no longer efficient[6]. The previous methods include sequential translation, parallel authoring and incremental just-in-time translation. Sequential translation is a method in which knowledge multilingualization occurs after completing source sentences. Parallel authoring is a method in which multilingual sentences are created at the same time by users in different languages. Incremental just-in-time translation is a method in which source sentences are created in a pre-defined language and are automatically translated by machine translators to be viewed by readers.

However, these methods are not useful for the Wiki community for particular reasons. In the sequential authoring method, original sentences should be completed before translation, but the original sentences in the Wiki are always changing. The multilingualization method in Wikipedia is accomplished by parallel authoring. However, there is a problem with the difference in the number of articles in each language version. In the incremental just-in-time translation, the system restricts writing to a pre-defined language set and there is a possibility that users' writing activities are inhibited. Because of these problems, a new method of multilingualization in the Wiki system is needed. Via research, a new Wiki system has been developed which enables users to visually distinguish what parts of contents are not yet translated. If users add original contents to a page, the system reminds that other pages still have parts which are not translated. However, the research supposes that all users are bilingual and that they can edit and translate articles immediately. Since this situation is not feasible for real Wikipedia translation community, this proposed system does not solve problems found in Wikipedia translations.

There are other systems used to support Wikipedia Translations such as Google Translator Toolkit and WikiBABEL[3]. These tools enable users to translate articles via machine translators, modify translation results and upload them to Wikipedia. WikiBABEL, developed by Microsoft Research, has a feature in which the system keeps layouts of target articles when users translate and edit them. The WikiBABEL has many dictionaries such as translation dictionary, parallel text dictionary and transliteration dictionary. But the Wik-

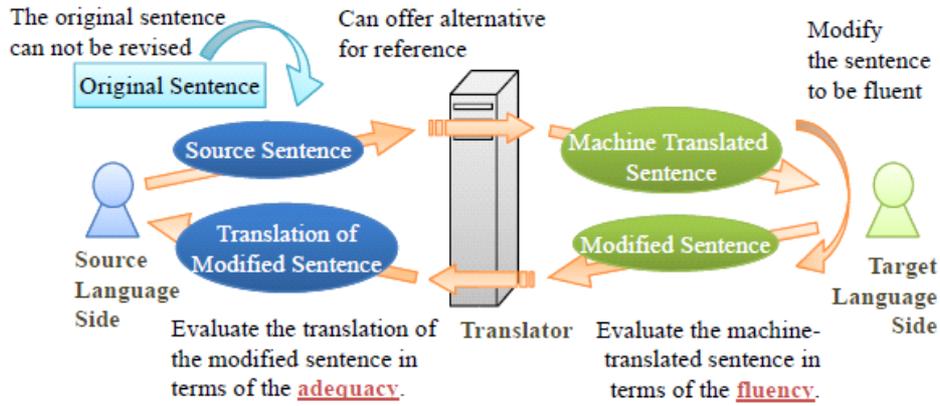


Figure 1: Collaborative translation protocol in Morita's research

iBABEL doesn't have discussion functions and it focuses on individual users' translation activity. On the other hand, the Google Translator Toolkit has discussion functions. In the discussion functions, users are able to discuss in real-time and share work spaces with pre-defined users who have exchanged their "Google accounts", since this function is realized through Google Chat. However, these discussions are monolingual communications and are not open like the Wikipedia community. Moreover, the translation activities are outside of Wikipedia. In this way, the system is not appropriate for the Wikipedia community where users place special emphasis on collaborations with others through revision history and discussion pages.

As mentioned above, previous tools for Wikipedia Translations serve as article translation functions but do not have appropriate discussion functions for translations of highly culture-specific articles. In this way, a discussion system for Wikipedia translations should exist for the "Wikipedia ways".

2.2 Collaborative Translations by Monolingual Users

In this research, the author has made an assumption that participants of Wikipedia translations are monolingual, middle-bilingual and bilingual users. In these Wikipedia translations, participants collaborate to translate articles. There is some previous research of collaborative translations. In this subsection, the author introduces two research activities which investigate collaborative translations among monolingual users.

D. Morita et al proposed a collaborative translation protocol which enables monolingual users to translate sentences by transferring the original sentences through machine translations[7]. Figure 1 shows the concept image of the collaborative translation protocol. First, the source language side user transfers an original sentence to the target language side user through a machine translator. The original sentence is automatically translated into a chosen target language. Then, the target language side user checks the fluency and modifies the translated sentence if needed. After the modification, the target language side user transfers the modified sentence to the source language side user through a machine translator. The modified sentence is automatically translated into the source language. Then the source language side user checks the accuracy by comparing machine translation results of the modified sentence with the original sentence. If the accuracy is good enough, the translation result of the original sentence is accepted. If it is not, the source language side user revises the original sentence and transfers it to the target language side user. In this protocol, users repeat this process until a good translation result is obtained. In Morita’s experiment, the protocol created high quality translation results compared with individual translations by using back-translations.

H. Chang et al developed the MonoTrans system which enables monolingual users to translate through machine translation mediated communication[8, 9]. MonoTrans’s communication is similar to Morita’s collaborative translation protocol. In this system, users can view posted messages and translation results in their own language. The system enables users to commit translation results of parts of sentences which are good enough, and to insert images into messages for visual communication with other language side users and to manage target sentences by progress lists. System messages are made by using predefined formats and parallel texts, so users can have high quality translations of messages. In the evaluation, users can create highly accurate and fluent translations compared to Google translation results.

However, the previous research does not deal with highly culture-specific sentences. Morita’s protocol targets general sentences and MonoTrans targets sentences in picture story books. In Wikipedia articles, sentences are written

in high quality in terms of "NPOV" and translators should make translation results into high quality sentences. In this way, it is difficult to judge translation qualities by checking machine translated sentences of translation results. Moreover, the fact that highly specific words or sentences cannot be translated by machine translators and the features of the loosely coordinated users in the Wikipedia community must be considered.

2.3 Wikipedia Translations Experiment

As mentioned in section 2.2, previous tools for Wikipedia translations provide monolingual discussion functions. However, considering the need for collaborations between translators and knowledge experts in translating highly culture-specific sentences, there should be a limitation in communications of monolingual discussions. The author analyzed conversation logs of the previous Wikipedia translation experiments[10] in order to understand what kind of problems occurred in such situations.

Details of the experiment environment are as follows: There were seven participants in the experiment, as shown in table 1, who had different language abilities. The experiment was carried out twice. In the first experiment, participants carried out discussions through a mailing list, and were encouraged to write in English. The target article was a "Sumo" article in the Japanese Wikipedia. The time period for the experiment was one week. The work space for translations was the MediaWiki¹⁾, a Wiki system used in the Wikipedia. For article translation, Text translation function of the Language Grid Toolbox[11] was used, which is a content management system provided by Language Grid Project[12]. In the second experiment, participants carried out discussions through the multilingual BBS function of the Language Grid Toolbox. The multilingual BBS enabled users to post and view messages in their own language by translating posted messages. The target article was a "Judo" article in the Japanese Wikipedia. Again, the work space for translations was the MediaWiki.

¹⁾ <http://www.mediawiki.org/wiki/MediaWiki/>

Table 1: Participants

	Native Language	Other Language
A	Japanese	English (Little)
B	Japanese	English (Little)
C	Japanese	English (Well)
D	Chinese	English (Well), Japanese (Little)
E	Finish	English (Well), Japanese (Little)
F	French	English (Well), Japanese (Little)
G	Bosnian	English (Well), Arabic (Little)

The translation process of the experiments was as follows. First of all, the Japanese article was translated into English, Chinese, French and Finnish by machine translators. Then, experiment participants marked mistranslated words as specific words and carried out discussions to create their own translations of these specific words. The translations of the specific words were stored as entries of a multilingual dictionary. The original article was then translated by machine translators in combination with the created multilingual dictionary. In the last step, translation results of the original article were revised by participants through discussions. The results of experiments conversation logs contain 60 topics and 193 messages.

The author analyzed these conversation logs and found problems and unique phenomena. The problems of the experiments are as follows.

- Changing languages in messages:

In the first experiment, the participants were encouraged to use English in writing discussion messages. But Japanese participants changed their communication to Japanese in the middle of the experiment. This change was not agreed upon by all participants, rather silently accepted but other non-Japanese participants.

- Conversation interruptions:

During discussions, several topics were simultaneously promoted. The topics were updated by several loosely coordinated participants. Therefore, participants were not able to follow the whole progress of topics and some topics were left behind in the process. The author found several interruption patterns which are as follows:

- Interruptions caused by not knowing the current status:

Several topics were interrupted because users did not know the current status of the topic at that time. For example, participant A, the source language side participant, posted a message explaining the meaning of the original sentences and then participant B, the target language side participant, accepted the explanations and revised the translation results in the work space. As participant A was not aware of this revision since participant B did not post a message to report the revisions, participant A was waiting to modify the translation results by participant B, while participant B was waiting to check the accuracy of the revised sentences by participant A. Both participants were waiting and the conversation was interrupted.

- Interruptions caused by unclear protocol:

During the experiment, there was a pre-defined translation process, but the process did not reveal the details of actions. Therefore, there were discussion interruptions because participants did not know what they should do in those particular situations. For example, participant B, the target language side participant asked participant A, the source language side participant, the meaning of the original sentence. Participant A, then answered the meaning and participant B accepted and sent the revised translation result. In this situation, participant A should check the accuracy of the translation result. However, participant A was satisfied with the success of the explanation of the meaning of the original sentence and did not check the accuracy. The conversation was then interrupted and the translation result was not given.

- Errant agreements:

The agreement regarding the translation of a part of a sentence was misunderstood as the agreement regarding the whole sentence. For example, participant B, the target language side participant, asked someone to explain the meaning of a part of a sentence because he was not able to understand and translate it. Participant A, the source language side participant, explained the meaning of the part, and participant B accepted the given explanation. Participant B then modified only the translation of that part, and participant A checked the accuracy and accepted it. In this situation, there was only an agreement regarding the translation of a part of a sentence. However, participant B thought he understood the meaning of the whole original sentence because he already got the explanation of parts he was not able to understand. On the other hand, participant A thought his task was complete, by explaining the meaning and checking the accuracy. In this case, the conversation was finished while the accuracy of the whole sentence was not checked.

- Difficulties in checking the accuracy:

During the experiment, the source language side participants were required to check the accuracy of translation results. However, translation results were too sophisticated for source language side participants to check their accuracy. In addition, the source language side participants were not able to check the accuracy because highly culture-specific sentences were translated into sentences of a different culture.

As mentioned above, there are many problems in collaborative translations of Wikipedia articles caused by the difference in cultures, user language abilities and features of Wiki users such as loosely coordinated collaborations. In the following sections, the author developed a system to support collaborative translations which solves these mentioned problems.

Chapter 3 Supporting Multilingual Communications on the MediaWiki

As mentioned in section 2.3, there is a need for users to communicate in their own native languages and discuss translations of highly culture-specific articles. The author designed a multilingual communication system in MediaWiki, which is the Wiki system used by the Wikipedia. In order to design the system, the author added multilingual capabilities to LiquidThreads, which is a bulletin board system running on MediaWiki. The author also considered the behavior of Wiki users. In this section, the author will first of all introduce MediaWiki and LiquidThreads. Then, function requirements of the system will be discussed. Moreover, details of the Multilingual LiquidThreads designs will be introduced. Finally, the design of PageDict, the dictionary function of the Language Grid Extension, will be introduced.

3.1 MediaWiki and LiquidThreads

MediaWiki is the Wiki system which has been developed by the Wikimedia foundation. There is an open source community to develop and enrich MediaWiki, and many MediaWiki extensions have been developed and released. MediaWiki extensions are a kind of add-ons and extensions which allow users to customize looks and works of MediaWiki¹⁾.

LiquidThreads²⁾ is also an extension of MediaWiki. LiquidThreads is a bulletin board system which runs on MediaWiki, especially in the Talk page. In previous discussions on the Talk page, users have used Wiki expression grammar in order to read contents easily. However, it depends upon users whether they wish to use the grammar or not so, the talk page tends to be complicated and it is difficult to understand the progress of each discussion. In order to solve this problem, Andrew Garret et al, developed LiquidThreads. In LiquidThreads, a message is called a "thread", and a topic is called "threads". LiquidThreads enables users to easily view threads by using a nested structure, sort functions

¹⁾ <http://www.mediawiki.org/wiki/Manual:Extensions>

²⁾ <http://www.mediawiki.org/wiki/Extension:LiquidThreads>

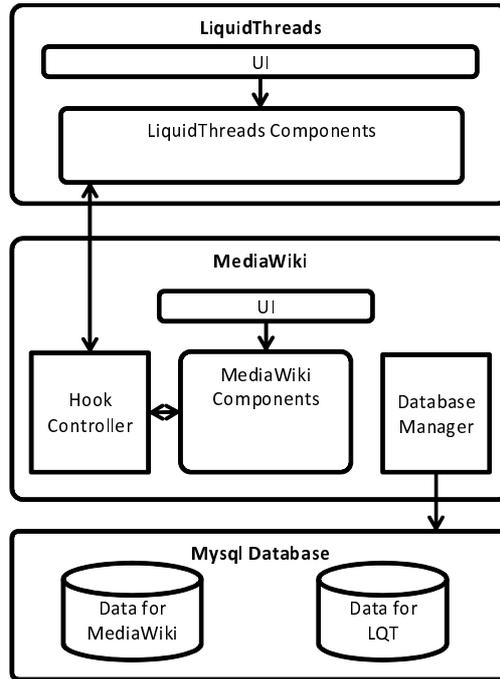


Figure 2: System architecture of LiquidThreads and MediaWiki

based on the date and link them to threads in TOC.

Figure 2 illustrates the system architecture of MediaWiki and LiquidThreads. MediaWiki contains basic data such as user profiles, page and revision histories. Moreover, developers can add tables in for extensions in the MediaWiki database. LiquidThreads contains data regarding threads in the database. Thread data manages message data but not the message body data. The message body data is stored as page data which is the basic data in MediaWiki to manage pages, and thread data has links to the page data. By using page data for thread body management, LiquidThreads enables users to edit threads and view revision history of these threads.

Details of different LiquidThreads and MediaWiki parts shown in figure 2 are as follows: LiquidThreads has a user interface and components. LiquidThreads components contain various functions in order to manage threads and user interface. MediaWiki also has components in order to manage pages, users and extensions. Moreover, MediaWiki contains user interfaces, hook controller and a database manager. The hook controller manages hooks included in the code of MediaWiki and other extensions. A hook is a label which is combined with

a function. When a hook appears in the code, the hook controller is called which further calls a function combined with the hook. Therefore, if developers change a function combined with a hook, they can easily change a function without changing any of the functions. Extensions are realized by using this hook system. The database manager enables functions to access a database with easy format requests. All functions of MediaWiki and extensions call the database manager when they access the database. The author uses the hook system to realize multilingual capabilities of LiquidThreads.

3.2 Functional Requirements

The author added multilingual capabilities to LiquidThreads in order to design a multilingual communication system. As with the previous multilingual communication systems, the author introduces the multilingual BBS of the Language Grid Toolbox and the multilingual Chat of the Language Grid Playground. Both of these systems translate messages by machine translators which use pre-defined languages. For example, if a community uses English, Chinese and French for discussions, the multilingual BBS translates messages into the other two languages in the posting phase. At the same time, users can modify machine translation results by referring to back-translations of messages to brush up translated results[13]. In this way, previous multilingual communication systems translate messages by using pre-defined languages.

However, it is difficult to use pre-defined languages for machine translators in Wikipedia translations because Wikipedia users have unspecified language abilities for translation collaborations. And, it is possible for other users to use new languages in a discussion. In this case, the pre-defined language model is not appropriate because the system should translate all messages into new languages. Therefore, the author designed the system which translates messages in the viewing phase rather than in the posting phase. In this way, the system can answer viewing requests in any language.

The LiquidThreads system was originally designed for monolingual communications. In order to add multilingual capabilities to LiquidThreads, the need to store translation results and distinguish the original language of messages

must be considered. This is because the system can translate messages into a requested language only when it knows what language was used to post the original message. Moreover, the author created an object "Translated Thread" in order to store translation results of a message. For multilingual capabilities, the system exchanges a thread with a translated thread which contains translated results in a requested language. If there is no appropriate translated thread of a thread, the system then translates the original message and creates a translated thread. The system can take a long time to translate threads, but the system will show data as quickly as possible by using stored data.

In order to access machine translators from MediaWiki, the author used the Language Grid Extension. The Language Grid Extension is an extension of MediaWiki which contains text translation, a multilingual dictionary combined with machine translators and a user interface of translation settings. Through the Language Grid Extension, users can access language services of the Language Grid which is a P2P language service grid. The Language Grid has many language services such as machine translators, parallel texts, morphological analyzers and multilingual dictionaries. For example, there are five English-Japanese machine translators in the Language Grid. Machine translators have fields and domains for sentences, translating them with high quality. One of the many advantages of the Language Grid is that users can select the most appropriate language service for their tasks. The Language Grid Extension enables users to easily change settings with a simple user interface.

Considering highly culture-specific articles translations and translation discussions, each page should have translation settings. Moreover, in machine translation mediated communications, communications often suffer from inconsistency and asymmetry in machine translations[14]. To resolve this problem, machine translations are combined with a dictionary service provided by the Language Grid. This service extracts specific words in original sentences, which are present in different dictionaries, and once machine translations are complete, an exchange of translated results with translations of specific words stored in dictionaries is carried out. However, in Wikipedia articles sometimes the same words have different meanings in different articles. To resolve this problem,

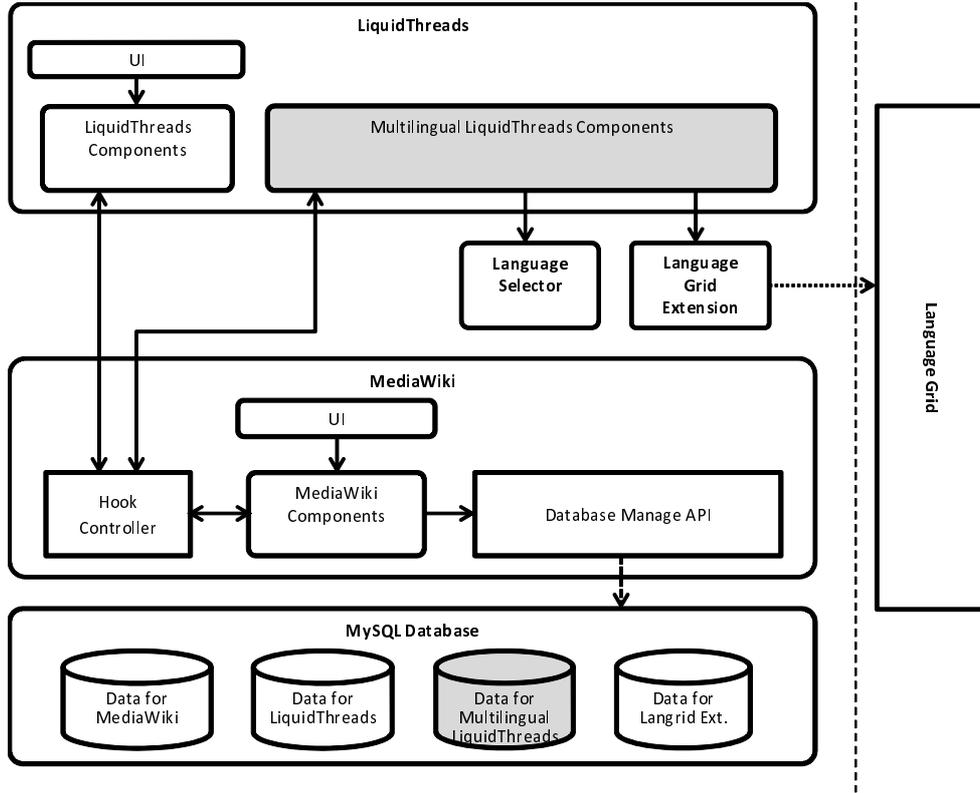


Figure 3: Whole architecture of Multilingual LiquidThreads

the author designed a dictionary function in the same way as the translation settings: each page has a dictionary. Moreover, page dictionaries in similar categories can be useful for translations of the target article. The author then designed import and export functions to use other page dictionaries for a particular page.

3.3 Multilingual LiquidThreads

This section introduces the details of Multilingual LiquidThreads which is an extension of the LiquidThreads developed by adding multilingual capabilities.

The system architecture of Multilingual LiquidThreads is as follows: Figure 3 illustrates the system architecture. As mentioned in section 3.1, there is hook system in MediaWiki. The previous hook controller in figure 2 calls the LiquidThreads and MediaWiki components. The author changed certain functions called by hooks with Multilingual LiquidThreads components. In this way, Multilingual LiquidThreads is developed by changing the code of LiquidThreads.

The Multilingual LiquidThreads components call two external extensions, the LanguageSelector¹⁾ and the Language Grid Extension. The LanguageSelector extension detects the interface language to be used by anonymous visitors, and provides a menu for language selection by users per page. Originally, the language of MediaWiki's interface could be changed because the MediaWiki project is a multilingual project. MediaWiki changes its language interface based on the language settings in the user profile. LanguageSelector enables users to change the language of MediaWiki's interface by selecting a language from the given choices in the language menu. LanguageSelector can access user profiles and overwrite their language settings. In this way, users can change the language of MediaWiki interface quite easily and automatically set language settings. Another external extension of the Language Grid Extension enables the access to various language services of the Language Grid. The author will introduce these details in section 3.4.

The viewing process of the Multilingual LiquidThreads will be shown in following paragraphs:

Figure 4 further illustrates the details of the system architecture which are related to the viewing process of the Multilingual LiquidThreads. In the Multilingual LiquidThreads components, the Multilingual Viewer has all the functions which are called from the hook controller. Multilingual Viewer calls several other components of the Multilingual LiquidThreads components based on the request from the hook controller. For example, the Thread Language Manager is called by the Multilingual Viewer when an original message is posted, in order to store the source language of the message. The Translated Thread Manager is called to create or load the translated message. And, there are components to access the external extensions such as the LanguageSelector Invocator and the Langrid Access Object. The Multilingual LiquidThreads components have two tables in the database such as the Thread Language and the Translated Thread. The Thread Language Manager and the Translated Thread Manager manage each table. They access these tables through the database manager in Medi-

¹⁾ <http://www.mediawiki.org/wiki/Extension:LanguageSelector>

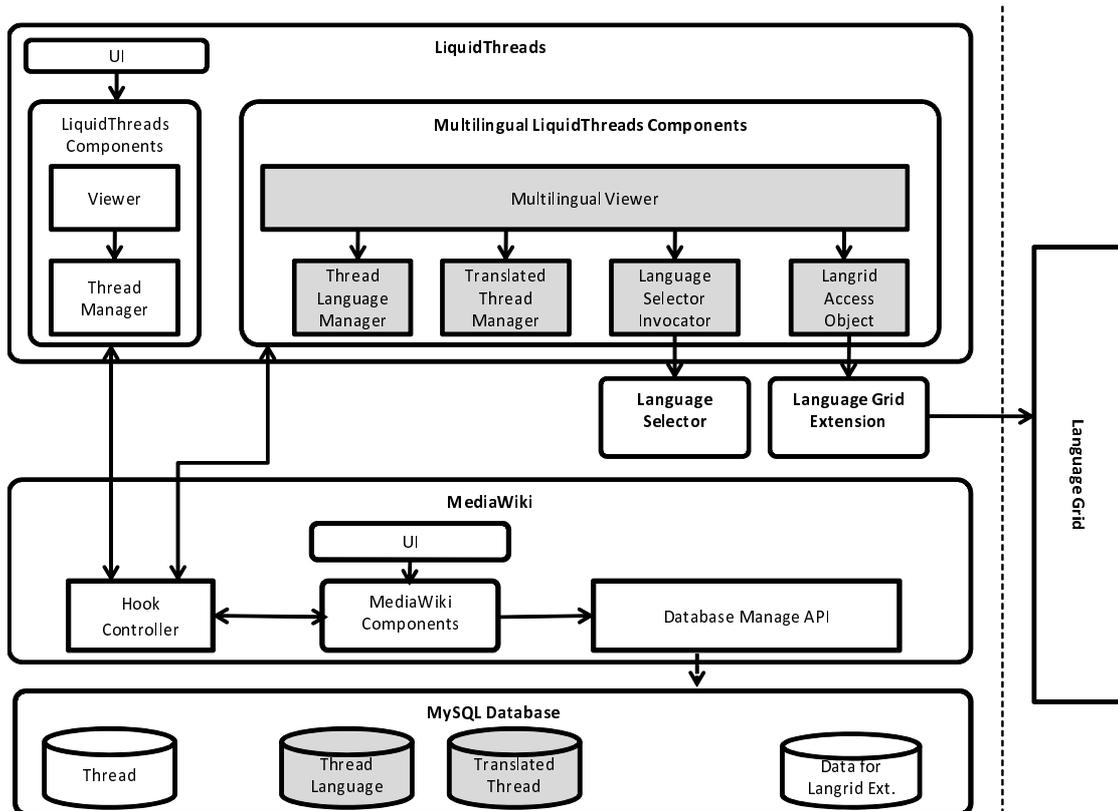


Figure 4: Part of system architecture for multilingual viewing

aWiki. In the Thread Language table, the source languages of the messages are stored. By using the table, the system can judge whether they should show the thread or the translated thread. The Translated Thread table has data to manage translated messages in the same way as the Thread table in LiquidThreads, to manage original messages. Multilingual LiquidThreads translates messages into requested languages and stores them into the Translated Thread table.

Figure 5 represents data structures of the LiquidThreads and the Multilingual LiquidThreads. The "thread" is an object of the LiquidThreads to manage message data. The thread has thread_id, thread_root, thread_subject and thread_parent. The thread_id is an id attached to all threads to distinguish them. The thread_subject has a subject of the topic which includes the thread. The thread_parent is a thread_id of the target thread for replying. The thread_root has id of a page object which manages the message body. In this way, a thread object does not have message body text data. The text data is

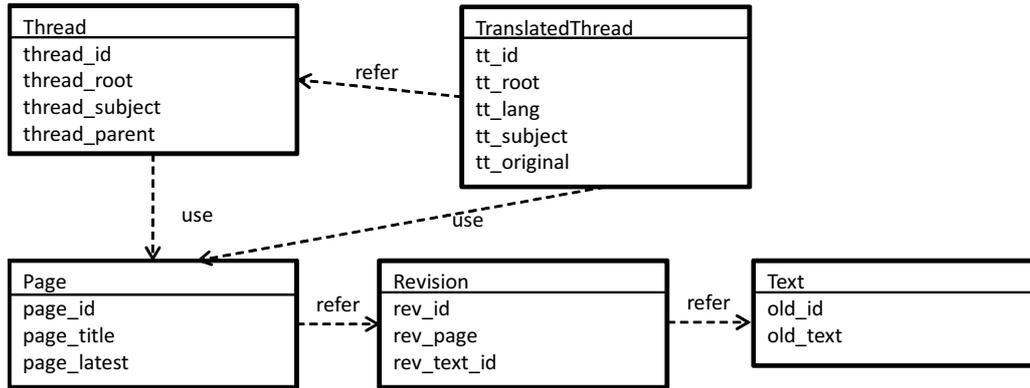


Figure 5: Data structure

stored in the page object. Because the page object has an editing function and a revision history, the thread uses the page object to manage the message body.

Page data manages texts in the page and revision history of MediaWiki, but the data is divided into three objects; page, text and revision. The page object has a page_title and page_latest. The page_title is the title of a page. The page_latest is id of the latest data revision. The revision object has all of the revision data of all pages. A page shows the latest version of the page, but the object revision enables users to view the past versions of a page. If there are several revisions of a page, the latest revisions will be shown on the page first with links to other versions shown on the history page. A revision object has a link to a text object. In MediaWiki, the page object does not have text data while the revision object has it because every page should be edited by many users. If the page object had text data which would be overwritten during the editing phase, MediaWiki would not be able to restore previous versions. In this way, pages are managed by three objects in MediaWiki and LiquidThreads using this method to manage message data.

The Translated Thread object manages translated results of messages, as well as the Thread object in LiquidThreads. The Translated Thread contains tt_id, tt_root, tt_lang, tt_subject, tt_original. The tt_id is an id attached to all translated threads in order to distinguish them. The tt_root has the id of a page object which manages the translated result body as well as the thread_id in Thread object. In this way, the system enables users to use the editing function

and revision history for translated messages. These functions are called "post editing" of translated messages. The post editing function enables users to revise machine translation results of messages and have smooth multilingual discussions. The Translated Thread object has `tt_lang` data for the language of the translated message, `tt_subject` data for translated results of the subject and `tt_original` data for the link to a thread which manages the original message.

Figure 6 illustrates the process of the multilingual viewing phase as a sequence diagram. The viewing process will be explained here. In the first step, a user selects a language from the LanguageSelector menu in the LiquidThreads user interface. Then, the UI calls the Viewer for illustrating the UI in the selected language with the title of the page. With the page title, the Viewer calls the Thread Manager to get all threads which are contained in the page. The threads are returned as an array of the thread. Then, the Hook controller is called by a hook named "showThreadContent." In the process of showThreadContent, threads in the array are illustrated separately. In this process, the Multilingual LiquidThreads exchanges original messages with translated ones based on the requested language.

The showThreadContent process is performed recursively for each thread in the array. The Hook Controller calls the Multilingual Viewer with a thread. The Multilingual Viewer gets the source language of the thread from the Thread Language Manager based on the thread id. Then the Multilingual Viewer gets the requested language from Language Selector Invocator. The Multilingual Viewer compares the source language and the requested language to judge whether there is a need to exchange the thread with an appropriate translated thread. If the source language and the requested language are the same, the thread should be illustrated and the Multilingual Viewer returns the page which manages the original message based on the `thread_root` data in the thread.

On the other hand, if these two languages are different, the Multilingual Viewer exchanges the thread with a translated one. The Multilingual Viewer calls the Translated Thread Manager with the thread id and the requested language in order to get the appropriate translated thread of the original thread. The Translated Thread Manager returns the translated thread if there is one.

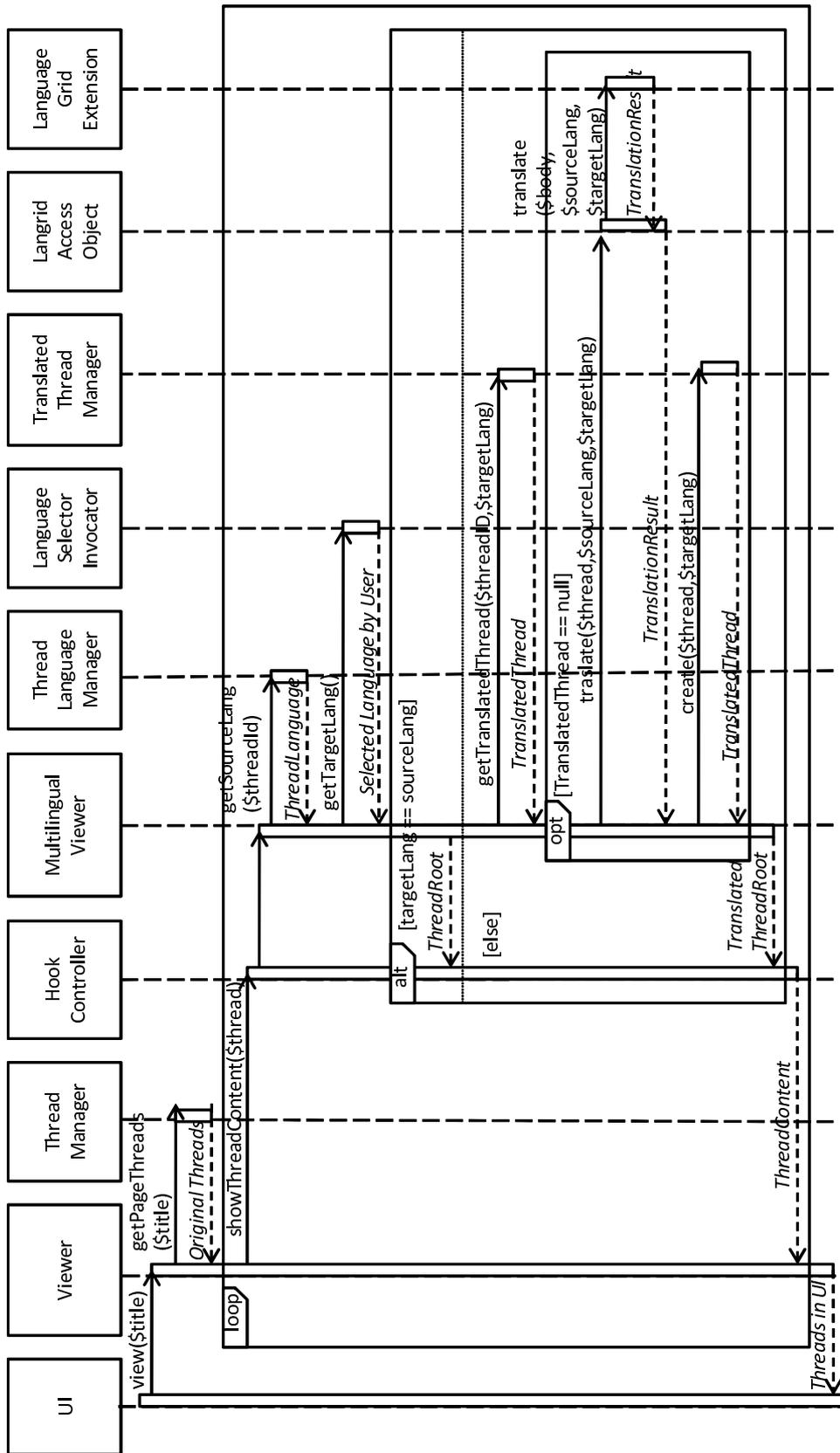


Figure 6: Viewing process of Multilingual LiquidThreads

If there is no appropriate translated thread, the Translated Thread Manager returns "null." When the Multilingual Viewer receives "null" from the Translated Thread Manager, the Multilingual Viewer calls the Langrid Access Object to create a translated thread by translating the original thread. The Langrid Access Object translates the message body by using language services of the Language Grid through the Language Grid Extension. Then, the Langrid Access Object returns translation results and the Multilingual Viewer creates a translated thread by calling the Translated Thread Manager. Finally, the Translated Thread Manager returns the appropriate translated thread to the Multilingual Viewer. At this moment, Multilingual Viewer receives the appropriate translated thread, and returns the page which manages the translated message based on the `tt_root` data in the translated thread. In this way, a page is returned from the Multilingual Viewer regardless whether the source language and the requested language are the same or not. Therefore, the Viewer can illustrate a message in both cases. This process is done for all threads in the array. In this way, the Multilingual LiquidThreads illustrates all messages in the requested language.

Figure 7 illustrates the process of the Multilingual LiquidThreads posting phase as a sequence diagram. The posting process is explained with this diagram: In the first step, a user clicks the save button in the posting form. Then, the post saving process is invoked. The invocation parameters are the message body and the posted language, which the user selected from the LanguageSelector menu. The Viewer then calls the Thread Manager to create a thread object for the message. The Thread Manager creates the thread and returns the thread id to the Viewer. The Hook Controller is then called by the hook named "AfterNewPostMetadataUpdate". In the AfterNewPostMetadataUpdate process, the source language data is saved for multilingual capabilities. The Hook Controller calls the Multilingual Viewer with the thread id and the posted language, and the Multilingual Viewer calls the Thread Language Manager to create a thread language object in order to save the source language. The Thread Language Manager returns "true" after successfully creating a thread language object. The "true" message is returned from the Multilingual Viewer

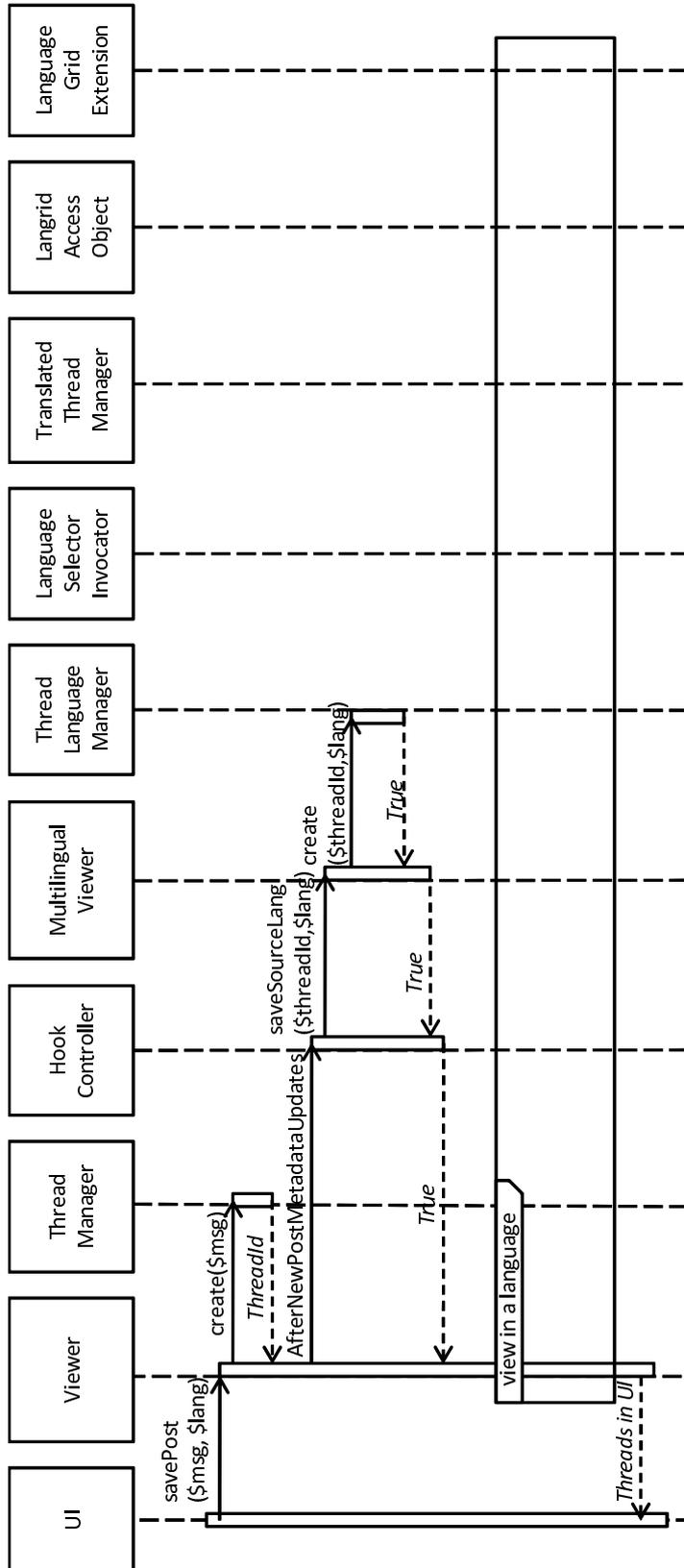


Figure 7: Posting process of Multilingual LiquidThreads

Tendenbarabara

[History](#) [Summarize](#) [Change subject](#)

What does Tendenbarabara mean?
Here is the original context:
But **Tendenbarabara** the temples and shrines along the railway company in the Kansai Ehou this year is ◦ ◦ for beginning to promote and,....

ParticipantB 08:14, 20 January 2011 Posted in English ← **Posted language** [Reply](#) [More](#) ▾

Tendenbarabara and I think that people act differently. Tendenbarabara sentence meaning are Edited by 0 users.
included, roughly, is that it depends on Ehou shrine that was advertised.

Original sentence
Tendenbarabaraとは、みんなが異なる行動をすることだと思います。Tendenbarabaraが含まれている文の意味は、だいたい、神社によって異なるEhouが宣伝されたということです。

[Hide](#)
ParticipantD 09:45, 20 January 2011 Posted in Japanese [Reply](#) [Parent](#) [Original](#) [More](#) ▾

Is Tendenbarabara something referring to the onomatopoeia word for scattering? I have never heard a "barabara" word in a different context. If this is the case, what does "tenden" mean?

ParticipantA 02:27, 21 January 2011 Posted in English [Reply](#) [Parent](#) [More](#) ▾

"barabara" means scattering or lack of unity. "tenden" means the individual or each person.
So "tendenbarabara" indicates that each person act differently and there is no unity.

ParticipantC 13:41, 21 January 2011 Posted in Japanese [Reply](#) [Parent](#) [Original](#) [More](#) ▾

Figure 8: User interface of Multilingual LiquidThreads

and the Hook Controller in that order. Finally, when the Viewer gets the "true" message, the Viewer calls the process of the multilingual viewing phase which is described above in figure 6. In this way, the Multilingual LiquidThreads saves data and illustrates the user interface reflecting posted messages.

Figure 8 illustrates the user interface of the Multilingual LiquidThreads. Each message is shown by a rectangle and contains a message, user name, date and the posted language. If a message is a translated one, then a link "Original" is shown by the rectangle to show the original message, and if a user clicks the link, the original message is shown by the highlighted rectangle. By using subjects of translated threads, the TOC is also illustrated in the requested language.

In this way, the author designed the Multilingual LiquidThreads. The multilingual LiquidThreads was developed and released as the extension of the MediaWiki and used for multilingual communications such as the Wikipedia Translation Experiment. In this Wikipedia Translation Experiment, users translated

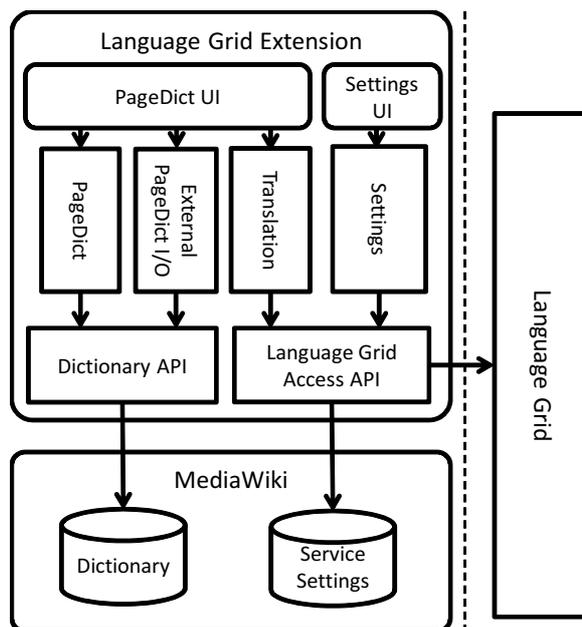


Figure 9: System architecture of Language Grid Extension

an English Wikipedia article on the Glacier National Park through multilingual discussions.

3.4 PageDict

Considering translations of highly culture-specific Wikipedia articles and translation discussions, each page should have translation settings. Moreover, sometimes the same words have different meanings in different Wikipedia articles. To resolve these problems, the author designed the PageDict function, which is dictionary function deployed for each page. The PageDict only influences the page and translation settings. The PageDict function is developed as one of the functions of the Language Grid Extension[?]. In this section, the author introduces the details of the Language Grid Extension.

The Language Grid Extension is an extension of MediaWiki with the purpose of providing access to different language services of the Language Grid. Figure 9 illustrates the system architecture of the Language Grid Extension. The Language Grid Extension has two user interfaces and four functions. The first page and function is the translation settings. In this page, users can select appropriate translators and combine them with various dictionaries through an



Figure 10: User interface of PageDict

easy and understandable interface. The second page and function is the PageDict. The PageDict is a dictionary function. In this page, users can add specific words and translations by editing the multilingual dictionary, and can automatically combine them with machine translations. This function was developed by using the dictionary API provided by the Language Grid Toolbox[11].

The third page is the import and export function of the PageDict. In Wikipedia, there is a category concept. In this concept, each page has some categories and pages of the same category have similar topics. The author has focused on these categories and has designed the import and export function of the PageDict. When users import other Pagedict, users are required to input the URL of the article. If users want to delete imported PageDict, they need to click the delete button in the list of imported PageDict. These functions enable consistency of translation results of specific words and discussions.

Figure 10 shows the interface of PageDict. In PageDict, there are two columns in two languages. Users can freely add over two languages, but the interface shows only two languages at a given time. This is why users edit pairs of original words and translations at a given time, so the need for displaying only two languages exists. Users can insert a new row by clicking the "add

record” link at the upper end of the table. Cells in the new row are blank, and users can add original words and translations in cells. Then, users click the ”save” button, to save their changes. The languages shown in two columns are changed by selecting appropriate languages from the menu on the upper end of each column. Users can search entries by using the search box placed at the top of the PageDict interface.

The PageDict function is developed as one of the functions of the Language Grid Extension, and the Language Grid Extension is opened as an open source software.

Chapter 4 Collaborative Translation Protocol for Wikipedia Translations

As mentioned in section 2.3, there are problems caused by unclear protocols for Wikipedia Translations. To resolve the problems, the author proposes a protocol for collaborative translations of Wikipedia articles. The author calls this protocol the "Collaborative Translation Protocol." In this section, the author introduces a previous research, "The Coordinator", which controls users' collaborations based on a protocol. Then, the process of creating the collaborative translation protocol is explained. Finally, the author explains the details of the protocol and validates the evidence of the protocol.

4.1 Controlling Users' Actions based on the Protocol

T. Winograd et al., developed "The Coordinator" system which was a groupware system that controlled users' collaborations based on a protocol[15]. The Coordinator supported communications of e-mails. It attached a tag representing utterance intensions to clarify the intensions of the e-mail, analyzed the state and controlled the users' collaborations.

Figure 11 illustrates a protocol used by the Coordinator. This is a protocol for the "conversation for action." There are two users in the protocol. The first user is the initial speaker (A) and the second user is the hearer (B). The Initial state is State 1, in which the initial speaker sends a request. Then, the hearer can select an action from three choices such as "Promise," "Counter" and "Reject." If the hearer promises to do the request, then the state is transited to state 3. If the hearer counters the request, then the state is transited to state 6. If the hearer rejects the request, then the state is transited to state 8 and the conversation is finished because state 8 is an end state. In this way, users' actions are controlled based on the states.

The Coordinator was extended to the real world as a groupware system, and a lot of companies used the system. There was much praise and critique towards the system. For example, Suchman criticized that attaching tags to the message was difficult for users and it was possible that the contents of the mails

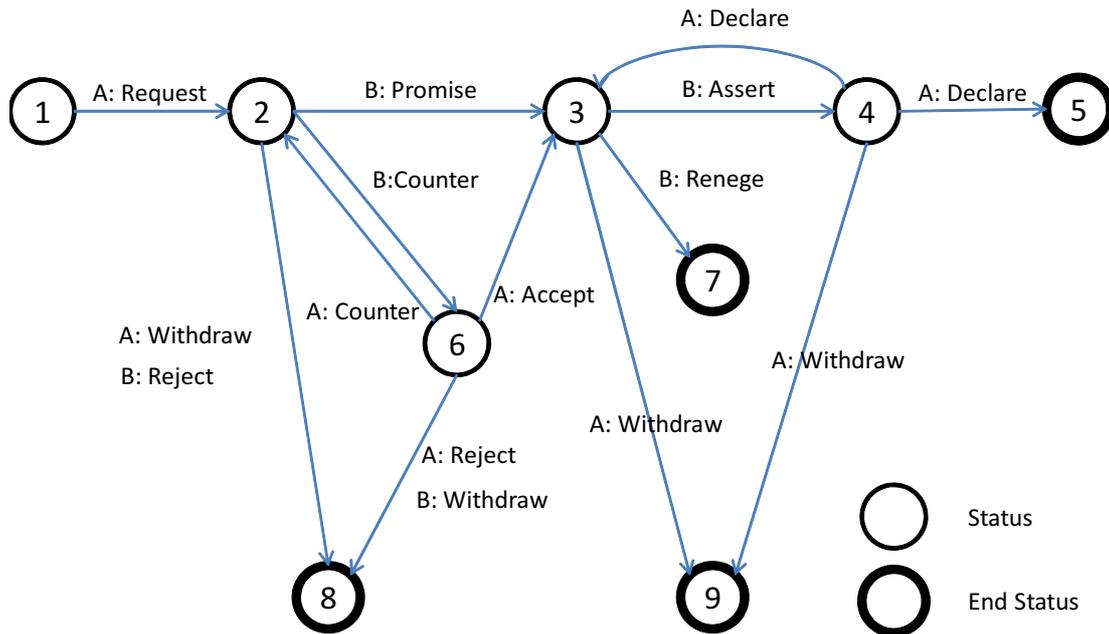


Figure 11: State transition diagram of the conversation for action

were restricted by attaching tags[16]. Bullen et al., pointed out there were real cases in which users ignored attaching appropriate tags when they used The Coordinator system[17]. On the other hand, Johnson et al., did the collaboration experiments in several conditions and reported that the usefulness of the Coordinator system was low in the condition that distances of the participants were short and they could discuss face-to-face, but that usefulness was high in the situation that there were long distances between the participants[18].

In this research, the author uses the same method as The Coordinator in order to control users' collaborations in Wikipedia translations. This method will be useful because there are long distances and time lags between the collaborators. In fact, the experiment introduces, in the section 2.3, an incident in which several discussions went on in parallel, and users missed the progress of the conversations and what they should do at that time. As the result, some discussions were left behind. Therefore, it is important to create a collaborative translation protocol and to control users' collaborations in Wikipedia translations based on the protocol.

Figure 12 illustrates the state transition diagram of Morita's protocol. This

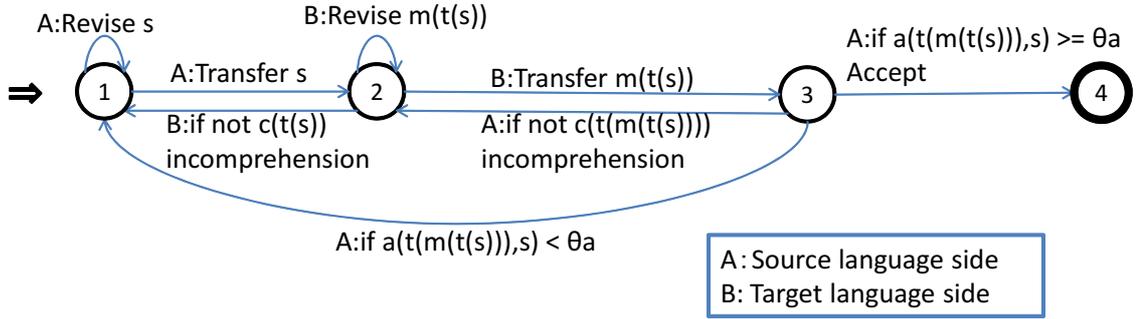


Figure 12: State transition diagram of the Morita protocol

state transition diagram has four states without end status of finishing collaborations with failures. The initial state is state 1, which is a state of "waiting to transfer original sentences." The source language side users transfer the original sentences. This action transits the state to state 2, which is a state of "waiting to modify the translated sentences." Then the target language side user modifies and transfers the modified sentences to the source language side user. This action transits the state to state 3, which is a state of "waiting to check accuracy." The source language side user checks the accuracy of the modified sentences. If the user accepts the translated result, then the state is transited to state 4. The state 4 is an end state of "finishing creating the translations successfully." If the source language side user rejects the translated results, then the state is transited to state 1 in order to ask the source language side user to revise the original sentences. If the source language side users cannot understand the machine translation results of the modified sentences, then the user asks the target language side user to modify the translated result again. This action transits the state to state 2. In this way, the state transition diagram of the Morita's protocol enables users to clear the actions of any users' situations.

4.2 Designing the Protocol

The author designed a collaborative translation protocol for Wikipedia translations. In this section, the details of the creation of the protocol are explained. This protocols targets collaborations between the source language side users and the target language side users to create a translation of one original sen-

Table 2: Tags representing utterance intentions in The Coordinator

Category	Tag
Assertives	
Directives	Request, Counter
Commissives	Promise, Accept, Reject, Withdraw, Renege
Expressives	
Declarations	Declare

tence. In this protocol, users ask questions, answer the questions and report their modifications of original/translated sentences with utterances.

For the first step, the author created tags to clear utterance intensions. In the second step, states of the collaborative translation protocol are designed. Then the tags attached to conversation logs in the third step. In the fourth step, state transition diagrams are created for each conversation log. In the final step, those state transition diagrams are integrated and generalized to create one diagram. Finally, the state transition diagram representing the collaborative translation protocol is given.

For the first step, the author created tags representing utterance intentions. Winograd et al., used five categories of general verbs. The five categories are as follows; Assertives, Directives, Commissives, Expressives and Declarations. The category of Assertives includes tags to commit the speaker to something's being the case. The category of Directives includes tags to attempt to get the hearer to do something such as "questions" and "commands." The category of Commissives includes tags to commit the speaker to some feature course of action. The category of Expressives includes tags to express a psychological state about a state of affairs such as "apologizing" and "prazing." The category of Declarations includes tags to bring about the correspondence between the propositional content of the speech act and reality. Table 2 shows the classification of the tags used in the Winograd's protocol illustrated in figure 11.

Winograd used explicit performative verbs for tags representing utterance intensions in the protocol. However, these tags are too general to describe the details of discussions in the Wikipedia Translations. Therefore, the author

Table 3: Tags representing utterance intentions for collaborative translations

Category	Tag
Assertives	Report revisions of original/translated sentences, Explain original/translated sentences
Directives	Modify the translated sentences, Ask to modify the translated sentences, Ask to explain the original/translated sentences
Commissives	Accept, Reject, Withdraw
Expressives	
Declarations	

extends the tags to be more specialized in the Wikipedia translations. Table 3 shows the tags representing utterance intentions for collaborative translations. The author also uses the five categories as well as the Winograd’s classifications.

The category of the assertives includes ”report revisions of original/translated sentences” and ”explain original/translated sentences.” The tag of ”explain original/translated sentences” includes explaining meanings of the sentences, words and grammars, and modifying the original sentences to be more understandable. In the Morita’s protocol, there is the phase to modify the original sentences. In the Wikipedia translations, however, the original sentences are in the articles and users cannot modify the sentences just for translation purposes. Therefore, this action is regarded as a method to explain the original sentence meanings in the collaborative translation protocol. The tags ”explain the translated sentences” will appear when the target language side user explains the meaning of the translated sentences toward the source language side users who have to know but cannot understand them for checking accuracy. These utterance intentions are used for indicating users’ knowledge.

The category of the directives includes ”modify the translated sentences”, ”ask to modify the translated sentences” and ”ask to explain the original/translated sentences.” The tag ”ask to explain the original sentences” includes actions in which the target language side user makes a proposal of modifying the original sentences. The target language side user asks the source language side user to explain the original sentences by answering the yes-no answer proposals.

The category of the commissives includes "accept", "reject" and "withdraw". The tags of "accept" and "reject" will appear when the users evaluate the fluency/accuracy of the translated sentences. The tag of "withdraw" has never appeared in users' utterances, but when conversations are interrupted, the "withdraw" tags are attached. In the collaborative translation protocol, tags in the expressives and declarations categories are not used.

In the second step, the author designed the states of the protocol. The author designed the states based on the collaborative translation process. Moreover, the states in Morita's protocol are referenced. As shown in the figure 12, Morita's protocol has four states; "waiting to transfer original sentences," "waiting to modify translated sentences," "waiting to check accuracy of translated sentences" and "finishing creating the translations successfully." These states are in the case of collaborations among monolingual users to create translations. States should be changed to be appropriate for Wikipedia translations.

In Wikipedia translations, the translation targets are highly culturally specific articles. It is difficult to create only by transferring the original/translated sentences through machine translators as well as Morita's research method. In the Wikipedia translation experiment, the participants explained the meanings of the original/translated sentences. Moreover, there are some participants who had middle levels of language skill regarding the original/target languages, and they proposed the revisions of the translated results or original sentences. Regarding these observations, the author designed following six states; "waiting to modify translated sentences," "waiting to check accuracy of the translated sentences," "waiting to check fluency of translated sentences," "waiting to explain original sentences," "waiting to explain translated sentences," and "finishing creating the translations successfully."

Details of each state are explained as follows: The first state is "waiting to modify translated sentences." This state is a state when users are waiting to modify translated sentences stored in their work space. In the Morita's protocol, the initial state is "waiting to transfer original sentences" because the trigger of the protocol is the source language side user. However, the trigger of collaborative translation of Wikipedia articles is a target language side user who

finds some mistranslations and modifies them. So, the initial state is "waiting to modify translated sentences."

The second state is "waiting to check accuracy of the translated sentences." This state is a state when the target language side users modified the translated sentences and they are waiting to check accuracy of the translated sentences by the source language side users. This state exists also in the Morita's protocol. In the collaborative translations, it is important for users to agree with the other side users about the correctness of the translation result. When they agree, then the translation results of the original sentences are understood. So, the source language side users should check the accuracy of modified sentences.

The third state is "waiting to check fluency of translated sentences." This state is a state when the source language side user modified the translated sentences, and they are waiting to check fluency of the translated sentences by the target language users. In Wikipedia translations, there are many users who have various language abilities. Wikipedia provides a system to express their language abilities called "Babel¹." The Babel system use seven grades of language abilities; native, professional, near-native, advanced, intermediate, basic, and cannot understand. In such situations, there are many knowledge holders who use the source language natively but also have language ability of the target language. These users can modify the translated result. In this case, the accuracy of the modified sentences should be high by using back translations. However, the fluency should be checked by the target language side native user because there is no guarantee that the sentences are fluent enough. Therefore this state is needed.

In the Wikipedia translations, an article is translated into various languages with high quality expressions. Wikipedia is an encyclopedia on the Internet, and expressions in articles include abstruse terms. Therefore, there are situations in which users ask the other side users to explain original/translated sentences and the author created those states for the protocol.

The fourth state is "waiting to explain original sentences." This state ex-

¹) <http://en.wikipedia.org/wiki/Wikipedia:Babel>

ists when the target language side user asks to the other side user to explain the meanings of words or sentences in the original sentences. As mentioned above, the phase to modify the original sentences in the Morita's protocol is also included in this state.

The fifth state is "waiting to explain translated sentences." This state is a state when the source language side users ask the other side users to explain the meanings of the translated words/sentences in order to check their accuracy.

Finally, the sixth state is "finishing creating the translations successfully." This state is an end state. In this state, the source language side user agrees with the target language side users about the correctness of the translated result because the accuracy and fluency of the translated result should be guaranteed.

In this way, the author created the six states for the collaborative translation protocol. As the next step, the author attached tags to conversation logs of the Wikipedia translation experiment. A message is divided into several parts based on the utterance intentions and each part of message is attached with an appropriate tag.

Table 4 shows an example of attaching tags to the conversation log. In this conversation, the target language side, user E, and the source language side, user A, are discussing about a part of an original sentence "類似の格闘技の中では、特に常に前に出て押すことを重視するところに特徴がある。 (*Martial arts similar to sumo all have the characteristic of advancing while constantly pushing your opponent.*)". User A and E discussed about revising the original sentence. The author attached tags to this conversation as shown in table 4. In the first utterance, user E reported that he modified the translated results and asked someone to explain the part of the sentence which he could not understand. The utterance intention of the conversation was "ask to explain original sentences." Then user A replied with the revised expression of the part of the original sentences in order to explain the meanings of the part. The utterance intention was "explain original sentences." In the next utterance, user E translated the revised original sentence by using several machine translators and rejected the explanation because there were not any understandable results. Moreover, user E revised the original sentences by himself, translated in English and asked the

Table 4: An example of conversation log

	Messages	Tags
E	I corrected the fluency of the English translation result. However, there are still parts that are unclear in the current translation result. "There is a feature at the place where it's emphasized to go out before a constancy in particular in the martial arts of resemblance and push it." In this case the term, or a string of terms, that makes the sentence incomprehensible is - 特に常に前に出て (<i>Espacially, keep pussing and stepping forward</i>).	Ask to explain original sentence
A	原文を書き直すと、「類似している格闘技と比較した場合、相撲の特徴は”常に前にでて押すこと”を重視することである。」となります。これで一度英語に訳してもらえますか？ (<i>I revised the ariginal expressions as "By comparing to other similar martial arts, Sumo has a feature that it is important to 'keep stepping forward and pussing'." Could you tranalate this in English?</i>)	Explain the original sentences
E	「常に前にでて押すこと」これはまだ問題です。「常に」を Wiktionary で探したら、「Commonly」が出てきます。J-Server の結果は「a Constancy」。英語の結果は「To go out before a constancy and push it.」です。意味は全く分かりません。逆に、Google Translate の結果は「Always push to get out before」。この場合、Google Translate の結果のほうがよかったけど、100%意味が同じではないと思います。もし「常に、相手を前に出て押すこと」に書き直せば、結果は「Always pushing to get out before your opponent」。これは正しいですか？ (<i>"keep stepping forward and pussing" still includes problems. In the Wiktionary, translation of the "常に" is "Commonly." The tranalstion result of the J-Server is "a Constancy." The Translation result of the part of the sentence is "To go out before a constancy and push it." This is not understandable. By the Google Translator, the result is "Always push to get out before" and I think this is better but not perfect. If the original part can be revised as "常に、相手を前に出て押すこと," then the translation should be "Always pushing to get out before your opponent." Is this correct?</i>)	Reject, Ask to explain original sentence
A	英語翻訳結果が正しいかどうか、ちょっと僕には判断できません。原文を以下のようにしてはどうでしょうか？「常に前に進みながら、相手を押し続けること」 (<i>I cannot judge whether this translation result is good or not. Can you understand if I revise the original part as "Keep pushing your opponent while constantly advancing"?</i>)	Explain the original sentences
E	これはいいです「常に前に進みながら、相手を押し続けること」。英語を書きなおせば、「Keep pushing your opponent while constantly advancing」になります。それで正しいと思います。 (<i>"Keep pushing your opponent while constantly advancing" is good and I can traslate it. I think this is correct.</i>)	Accept

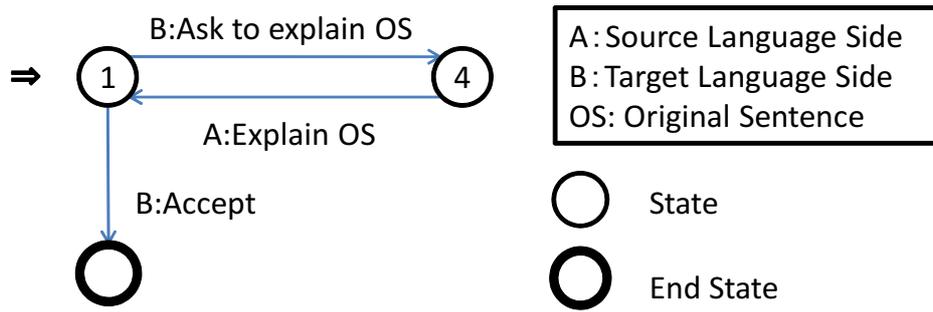


Figure 13: State transition diagram of the conversations shown in the table 4

target language side users to check the accuracy. In this situation, the author thought that user E asked user A to explain the original sentences by checking the correctness of the revised original sentence. Then, "reject" and "ask to explain original sentence" were attached to this utterance. User A rejected this proposal because he could not judge whether the translated result is correct or not. Then user A proposed the revised original sentence again. This utterance is attached the tags "reject" and "explain original sentence." Finally, user E accepted the revised version with the translation result of the part. "Accept" was attached to this utterance.

For the fourth step, state transition diagrams of each are created for each conversation with the states and attached tags. For the first step, the author divided the participants of the experiment, shown in the table 1, into two rolls; the source language side and the target language side. As the result, the source language side participants were A, B and C, and they should have specific cultural knowledge about "Sumo" and "Judo." On the other hand, the target language side participants were D, E, F and G, and they should have native language ability of the target language. The author described the state translation diagrams in terms of the conversations between these two rolls.

Figure 13 illustrates the state transition diagram of the conversation shown in the table 4. First, the target language side B asked the source language side A to explain the original sentence. This utterance transits the state to the state 4 which is "waiting to explain original sentences," Then, the source language side A explained the original sentence by the revising it. Then the state

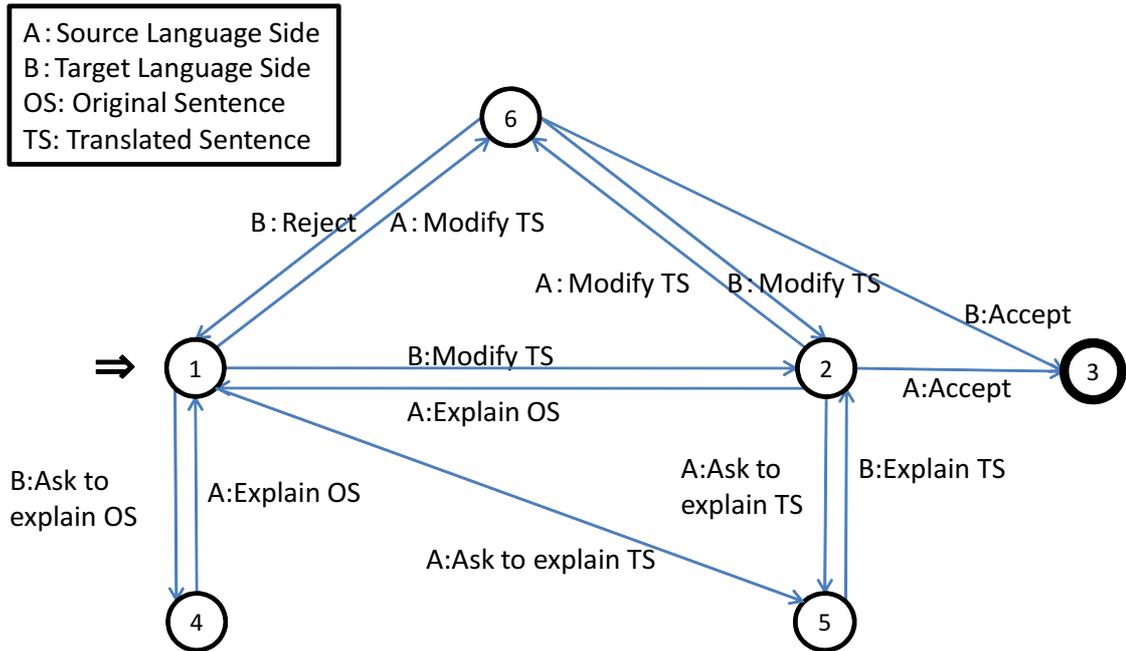


Figure 14: State transition diagram of collaborative translation protocol

returns to the initial state which is "waiting to modify translated sentences" because the target language side B will modify the translated sentence if he can understand the meaning of the original sentences. Then, the target language side B rejected and asked the source language side A to explain the original sentence by checking his proposal of the revision of the original sentence. The state is transited to the state 4 again. The source language side A rejected the proposal and explain the original sentences by revising it. And the state is transited to the state 1. Finally, the target language side B accepted and this conversation was finished.

In this way, the author created twelve state transition diagrams for the discussions to create the translated sentences. As the final step, the author merged them into one state transition diagram in terms of their state and generalized it. The "generalization" means that some transitions with utterance intensions, which did not appear in the conversation log of the experiment but there should be, are added.

4.3 Collaborative Translation Protocol

In this section, the author explains the collaborative transition protocol, which is the state transition diagram created by the process in the section 4.2.

Figure 14 illustrates the collaborative translation protocol. The states in the diagram are the states mentioned in the section 4.2; "(1) waiting to modify translated sentences," "(2) waiting to check accuracy of the translated sentences," "(3) waiting to check fluency of translated sentences," "(4) waiting to explain original sentences," "(5) waiting to explain translated sentences," "(6) finishing creating the translations successfully." The author explains the transitions from each state.

State 1 is "waiting to modify translated sentences." This is an initial state of the protocol. There are four transitions from this state. The first transition leaves for state 2 and occurs when the target language side user modifies the translated sentences. The second transition leaves for state 4 and occurs when the target language side user asks to explain the original sentences toward the source language side users. The third transition leaves for state 3 and occurs when the source language side user modifies the translated sentences. The fourth transition leaves for state 5 and occurs when the source language side user asks the target language side users to explain translated sentences in order to check whether the translation result should be modified.

State 2 is "waiting to check accuracy of the translated sentences." There are four transitions from this state. The first transition leaves for state 6 and occurs when the source language side user has accepted the translated sentences. The second transition leaves for state 1 and occurs when the source language side user rejects the translated sentences and explains the original sentence to give further explanations. The third transition leaves for state 3 and occurs when the source language side user rejects the translated sentences and modifies the translated sentences by himself. The fourth transition leaves for state 5 and occurs when the source language side user asks the target language side users to explain translated sentences in order to check accuracy.

State 3 is "waiting to check fluency of translated sentences." In this state, the accuracy of the translated sentences is guaranteed by the source language

side user. They can use back-translation function to check the accuracy of the translated sentence. There are three transitions from this state. The first transition leaves for state 6 and occurs when the target language side user accepted the fluency of the translated sentences. That's because the accuracy of the sentence is already checked by the source language side user. The second transition leaves for state 2 and occurs when the target language side user modifies the translated sentences to enrich its fluency. Then the translated sentences are changed and there is need for accuracy to be checked. The third transition leaves for state 1 and occurs when the target language side user rejects the translated sentences.

State 4 is "waiting to explain original sentences." This state has one transition leaving for the state 1 and occurs when the source language side user explains the original sentences. State 5 is a state of "waiting to explain translated sentences." This state has one transition leaving for state 2 and occurs when the target language side user explains the translated sentences.

State 6 is "finishing creating the translations successfully." This state is an end state with success of the collaborative translations. There should be a lot of end states with failure in the collaborative translations. They will occur when the discussions are interrupted by users' withdrawing. However, these interruptions can occur at the every state, so the author removes these states from the collaborative translation protocol.

4.4 Validation of the Protocol

As mentioned in section 2.3, there are a lot of problems in collaborative translations of the Wikipedia articles caused by difference of cultures, language abilities of users and the various features of Wiki users groups such as: loosely coordinated collaborations, interruption caused by unclear protocol, interruptions caused by missing progress and errant agreements. To resolve these problems, the author designed the collaborative translation protocol. In this section, the author validates the protocol by checking whether the protocol solves these problems.

First, the author validates the protocol about interruptions caused by un-

clear protocols. In the experiment, there was pre-defined process of translation, but the process did not reveal the details of the actions. Therefore there were interrupted discussions because participants did not know what they should do in those situations. In this protocol, there are six states and there are pre-defined actions at the each state. Therefore, users can know what they should do at that time. For example, the target language user modifies the translated sentences, then the state is "waiting to check accuracy," and the source language side users can know there is need to check the accuracy. In this way, the interruptions caused by unclear protocol will be prevented.

Second, the author validates the protocol about the interruptions caused by missing progress. The interruptions are situations when the both sides users are waiting for the other's actions and the conversation is interrupted. This interruption is caused because users don't know the current status of the topic at that time. In this protocol, each state of discussion is managed only by users' utterances. Then, users should say something if they do something. In the protocol, there is no epsilon transition, so users can follow the current state transited by previous utterances. In this way, the protocol forces users' utterance when they do something, and all users can know the current state. Then, the interruptions caused by missing progress will be prevented.

Third, the author validates the protocol about errant agreements. This is a situation when the agreement of the translation result about a part of the sentence is misunderstood as the agreement about the whole sentence. As mentioned in section 4.2, this protocol's targets collaborations between the source language side users and the target language side users to create a translation of one original sentence. In the example of the errant agreements in section 2.3, both sides misunderstood that they achieved agreement about an agreement about the translation result for part of the sentence only. In this protocol, agreements about the translated results of some parts of a sentence were agreements to accept explanations of the parts of original sentence. Moreover, in the example, they misunderstood that they had checked accuracy of the sentence, but in reality that action had been explained in the context of checking the accuracy of part of the sentence. They thought that the transitions of the states

were from state 1 to state 2, and then state 3. But in actuality, the transitions were from state 1 to state 2, and then back to state 1. In this way, the protocol will prevent errant agreements by restricting the target of the protocol as translations of one sentence.

Chapter 5 Supporting Multilingual Collaborations based on the Collaborative Translation Protocol

As mentioned in chapter Chapter 4, the author designed the collaborative translation protocol and validated it. In this chapter, development of the collaboration control system based on the protocol is discussed. The author explains the functional requirements of the system and the details of the implementations are presented.

5.1 Functional Requirements

As mentioned in section 4.4, the collaborative translation protocol can prevent many of the problems found in collaborative translations. The author developed the collaboration control system based on the protocol and integrated it into the Multilingual LiquidThreads.

In the collaborative translation protocol, it is important for the system to understand the users' utterance intensions because the protocol transits their states based on the utterance intensions. In order to understand, the system asks users to select their intensions from a menu of the possible intensions as well as The Coordinator. The possible intensions are created automatically by the system based on the protocol.

In order to control the users' collaborations on Multilingual LiquidThreads, the author designed a system in which each message has a tag expressing the utterance intension and a state in the protocol. When considering discussions in Wikipedia, there are many loosely coordinated users and they can break into discussions freely. If there is a discussion which is already finished, the users can resurrect it. Users can reply to messages whenever they want. Therefore each message should have a state to calculate the next desired reply state. Moreover, there are often several topics discussed in parallel, and users cannot follow all discussions' progress. To resolve this problem, the author designs the system which shows the most current states of each discussion at the table of contents.

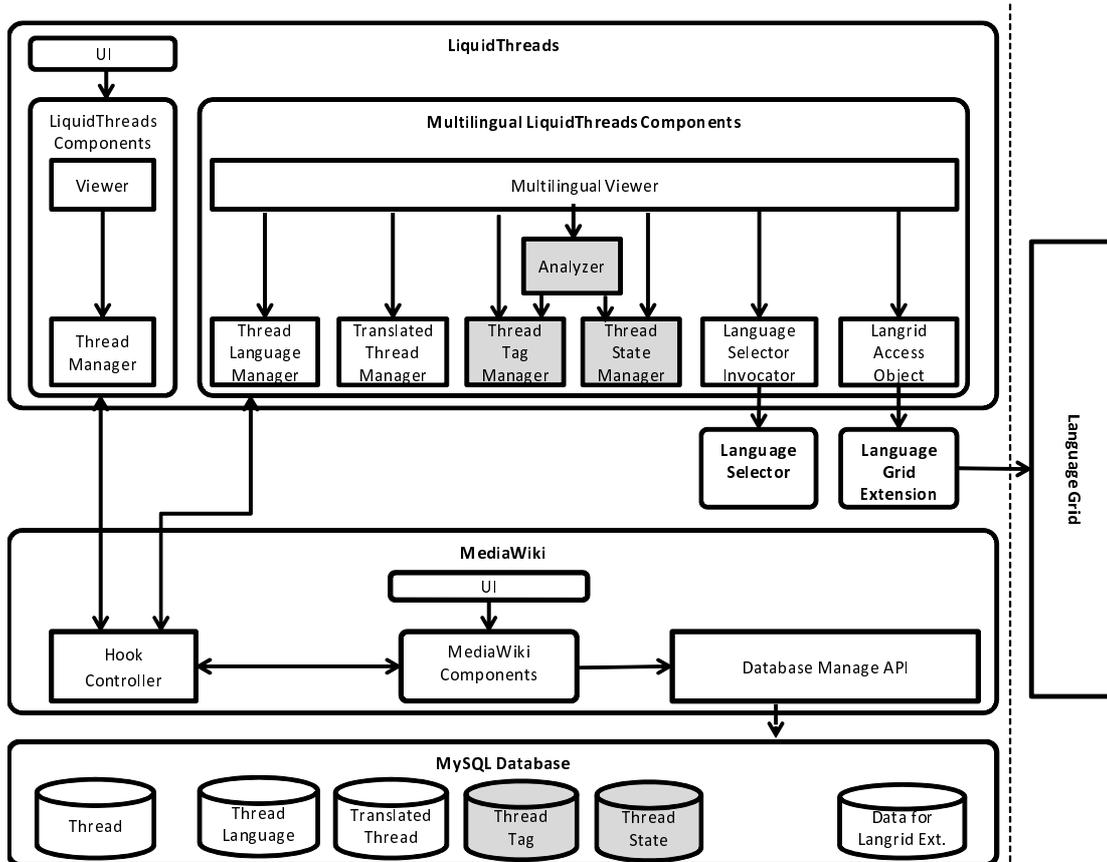


Figure 15: System architecture for controlling collaborations based on collaborative translation protocol

5.2 Implementation

This section introduces the details of the collaboration control system integrated into Multilingual LiquidThreads.

Figure 15 illustrates details of the system architecture of Multilingual LiquidThreads and Collaboration Control system. Three components and two data tables are added to the Multilingual LiquidThreads system shown in the figure 4. The three components are "Analyzer", "Thread Tag Manager" and "Thread State Manager" and the two data tables are "Thread Tag" and "Thread State".

The Analyzer calculates the next state and choices of the possible utterance intentions based on the protocol. The Thread Tag Manager manages the thread tag data in the Thread Tag table which contains utterance intentions of the messages. The Thread State Manager manages the Thread State data in the

Thread State table which contains states of the messages.

The collaborative translation protocol is implemented as follows: The Analyzer has the protocol data as an array. The array of the protocol contains ternary sets of "current state", "utterance intension" and "next state". A ternary set represents a transition.

The Analyzer creates the options for the possible utterance intentions. When a user opens the posting message form, the Analyzer creates the option of the utterance intensions based on the current state of the discussion. The current state discussion is given by the previous message's state. If the user starts new discussion, then the current state is the initial state. The Analyzer collects the ternary sets in the protocol array which "current state" are the same with the current state of the discussion. Then the utterance intensions in those ternary sets are regarded as the possible utterance intensions. In this way, the Analyzer can create the appropriate choices of the utterance intensions at that state.

Figure 16 illustrates the state calculating process in the posting phase of Multilingual LiquidThreads as a sequence diagram. In the posting phase of Multilingual Liquidthreads, shown in the figure 7, the user selects an original language of a message when saving the message. In this system, users are required to select an utterance intension in addition to the original language. When a user posts a message with a language and an utterance intension, then Multilingual LiquidThreads stores the message and language as thread and thread language data as explained in section 3.3. Then, the Multilingual Viewer calls Thread Tag Manager to store the utterance intension. The Thread Tag Manager returns a "true" message after creating the thread tag data successfully. Then the Multilingual Viewer calls the Analyzer to calculate the state of the posted message. In the calculating process, the Analyzer calls the Thread State manager firstly to get the previous state with the thread parent ID. The thread parent ID indicates the target thread for replying. So, the Analyzer can get the "previous" state of the discussion by getting the state of the parent thread. If there is no parent thread, then the "previous" state of the discussion is "initial state". Secondly, the Analyzer calculates the current state by searching the ternary set in the protocol array which contains the "previous"

state as the "current state" and the utterance intension of the message as the "utterance intension." If there is an appropriate set, then the Analyzer regards the "next state" in the set as the current state of the message. If there is no set, then the current state is "Unknown". Finally, the Analyzer calls the Thread State Manager to store the state with thread ID and the current state. Then the Thread State Manager creates the Thread State data and the calculating process is finished. After the calculating process, the Viewer calls the process of viewing in a selected language as well as the posting phase of Multilingual LiquidThreads.

Figure 17 and 18 illustrates the user interface of the Multilingual LiquidThreads with the collaboration control system. As shown in figure 17, an utterance intension and a state of a message are described in each rectangle of a message. Users can understand the utterance intensions and the state of the message visually. In the top of the talk page, there is a table of contents as shown in figure 18. The table of contents has the recent message state in addition to the thread title, number of replies and last modified date. Users can learn the progress of the all discussions by checking the recent message states in the table of contents.

In this way, the author developed the collaboration control system and integrated it with Multilingual LiquidThreads.

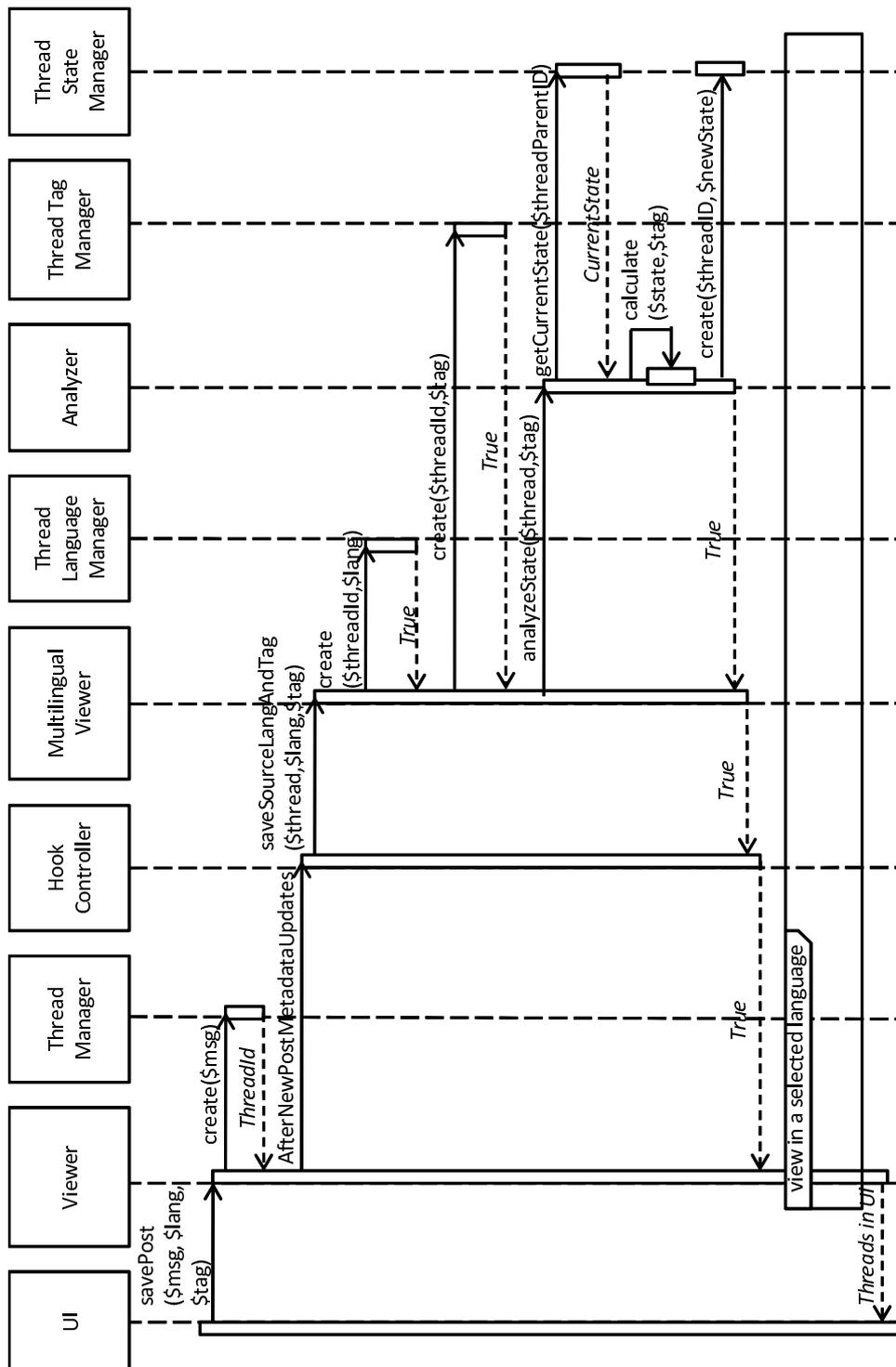


Figure 16: Controlling process based on the collaborative translation protocol

First Line: Modified.

Demon (Youkai), in the traditional Japanese folk beliefs, bizarre and unusual phenomena beyond human understanding, or cause them to be of an extraordinary existence with strange powers.

To

Demons (Youkai in Japanese), are found in traditional Japanese folklore. They represent the bizarre things beyond human understanding or the existence of extraordinary and strange powers.

ParticipantA 03:09, 26 January 2011 Posted in English Utterance: B:modify TS State: waiting to

I think good. To reflect the correction.

Utterance
Intention

State

ParticipantC 11:58, 27 January 2011 Posted in Japanese Utterance: A:accept State: Finish

Post accepted and modified.

ParticipantA 02:52, 28 January 2011 Posted in English Utterance: A:accept State: Finish

Figure 17: User interface of the system

Contents

Thread title	Replies	Last modified	Recent Message State
First Line: Modified.	2	03:08, 1 February 2011	Finish
God Tsukumo section kanji	4	07:25, 30 January 2011	waiting to modify TS
Overview: First part	3	07:23, 30 January 2011	Finish
Second line modified	3	03:26, 28 January 2011	Finish
Overview: Second sentence	0	15:06, 27 January 2011	waiting to check fluency

Figure 18: Showing status in TOC

Chapter 6 Internationally Expanding of the Multilingual LiquidThreads

In this research, the author developed the multilingual collaboration system to support Wikipedia translations in order to resolve the problem of imbalance among the number of Wikipedia articles in different languages. Multilingual LiquidThreads is an open-source software developed by extending LiquidThreads. Moreover, the Language Grid Extension, which offers translation and PageDict function for the Multilingual LiquidThreads, is also open source software. However, the Language Grid Extension has to access to the Language grid operated by the Graduate school of Informatics in Kyoto University. In the Language Grid at Kyoto University, a user account is needed to use the language services. To get an account, users must sign a "service grid agreement" created by the Kyoto operation center. Moreover, the purposes of using language services are restricted by the agreements and the terms of use. Users cannot access any language service through the Language Grid Extension without an account.

In order to use the language services from the Language grid Extension, it is important to establish a new Language Grid under the servers of Wikimedia Foundation. This grid is called the "Language Grid for Wikimedia." Then, Wikimedia Foundation can create agreements to operate the Language Grid for Wikimedia freely and users can use the language services under the original terms of use. In the Language Grid for Wikimedia Project, three software are being developed and will be released on the Wikimedia servers. The three software are Language Grid for Wikimedia, Language Grid Extension and Multilingual LiquidThreads. Wikimedia Foundation agreed with the National Institute of Information and Communication Technology (NICT) and the Graduate school of Informatics at Kyoto University to make the Language Grid for Wikimedia Project as a Wikimedia Foundation's research project.

Figure 19 illustrates the overview of the system architecture. The project provides three software; Language Grid for Wikimedia, Language Grid Extension and Multilingual LiquidThreads. These software are developed as open source software. On the Language Grid for Wikimedia, several language ser-

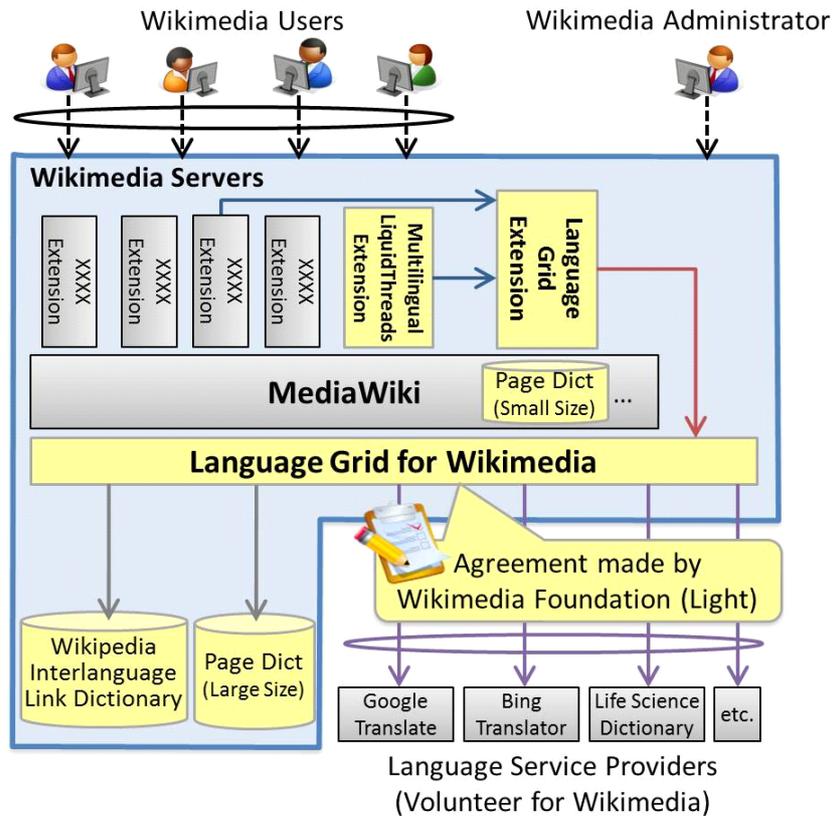


Figure 19: Language Grid for MediaWiki Project

vices are running, such as translators and dictionaries. These language services are provided from language service providers who sign the agreement made by Wikimedia Foundation. The Wikipedia Interlanguage dictionary and PageDict with large size will be created by users' activity, and users can access them through the Language Grid for Wikimedia. The Language Grid Extension and Multilingual LiquidThreads are also provided as extensions of MediaWiki. Users can use Multilingual LiquidThreads for the multilingual discussion and developers can create new multilingual services by using the Language Grid Extension to access the Language Grid.

Figure 20 illustrates the management system of the Language Grid for Wikimedia project. In the management system, project managers manage the developing teams of software. Multilingual LiquidThreads are developed and managed by the Graduate School of Informatics at Kyoto University. The Language Grid Extension and Language Grid for Wikimedia are developed and managed

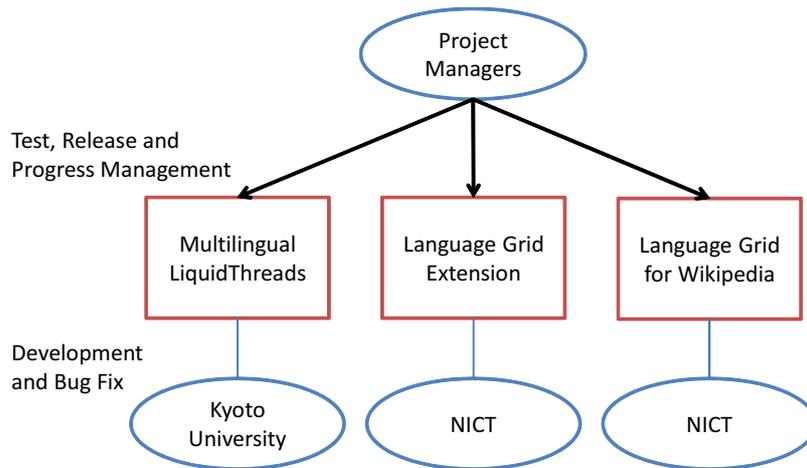


Figure 20: Manangement system

by the NICT. There will be several tests on the Wikimedia servers before operating the Language Grid for Wikimedia and the developers in Wikimedia Foundation will give feedback for the software. The Project Manager will manage developments and the application of these feedbacks through monthly meetings. Moreover, project managers manage the extension pages of the Language Grid Extension and Multilingual LiquidThreads, negotiate with developers in the Wikimedia open source community and run tests of software before their releases. Software managers will develop their software, fix bugs and manage the source code.

In this way, the Language Grid for Wikimedia aims to build multilingual communication environments and resolve the problem of imbalance among the number of Wikipedia articles in different languages.

Chapter 7 Conclusion

In this research, the author developed the multilingual environment on the MediaWiki to support multilingual collaborations for Wikipedia translations. Contributions of this research are as follows:

Developing multilingual BBS and a dictionary for each page and combining it with machine translation: The author designed the Multilingual LiquidThreads which is a multilingual BBS and translates messages in the viewing phase. The author also designed PageDict, which is a dictionary system combined with machine translations. These systems are opened as MediaWiki extensions.

Designing the collaborative translation protocol and the development of the collaboration control system based on the utterance intentions: The author designed the collaborative translation protocol by analyzing the logs of the collaborative translation experiment. This protocol is represented by a state transition diagram and transit state by utterance intentions. The author developed the collaboration control system based on the protocol and integrated it with the Multilingual LiquidThreads.

The future work will include management of the Language Grid for Wikimedia project to establish a multilingual environment in the Wikimedia community. The author will manage this project as a member of project managers. And the usability of the collaboration system will be evaluated through Wikipedia translation project. In the Wikipedia translation project, users who have various language abilities will translate Wikipedia articles in Japanese, Chinese, Korean and English et al. In the project, the Multilingual LiquidThreads with collaboration control system will be used for discussions of translations. The project managers will evaluate the system by observing conversations and users' ways of the system. The author is a member of the project managers and he will continue to contribute to the project.

Acknowledgments

The author would like to express sincere gratitude to the supervisor, Professor Toru Ishida at Kyoto University, for his continuous guidance, valuable advice, and helpful discussions.

The author would like to tender his acknowledgments to Associate Professor Shigeo Matsubara, Associate Professor David Kinny, Lecturer Rieko Inaba, Assistant Professor Hiromitsu Hattori and Assistant Professor Yuu Nakajima at Kyoto University, for their technical and constructive advice.

The author would like to express his appreciations to the advisers, Professor Masatoshi Yoshikawa and Associate Professor Yasuyuki Sumi at Kyoto University and Assistant Professor Satoshi Oyama at Hokkaido University for his valuable advice.

And the author is deeply grateful to Researcher Yohei Murakami at the NICT Language Grid Project, for his continuous advice and discussions.

Finally, the author would like to thank all members of Ishida and Matsubara laboratory for their various supports and discussions.

References

- [1] Desilets, A., Gonzalez, L., Paquet, S. and Stojanovic, M.: Becoming Wikipedians: Transformation of Participation in a Collaborative Online Encyclopedia, *Proceedings of GROUP International Conference on Supporting Group Work*, pp. 1–10 (2005).
- [2] Viegas, F. B., Wattenberg, M. and Dave, K.: Studying Cooperation and Conflict between Authors with History Flow Visualizations, *Proceedings of CHI'04*, pp. 575–582 (2004).
- [3] Kumaran, A., Saravanan, K., Datha, N., Ashok, B. and Dendi, V.: Wik-iBABEL: A Wiki-style Platform for Creation of Parallel Data, *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pp. 29–32 (2009).
- [4] Shahid, A. and Kazakov, D.: Automatic Multilingual Lexicon Generation using Wikipedia as a Resource, *International Conference on Agents and Artificial Intelligence* (2009).
- [5] Desilets, A.: Translation Wikified: How will massive online collaboration impact the world of translation?, *Keynote in Translating and the Computer* (2007).
- [6] Desilets, A., Gonzalez, S., Paquet, S. and Stojanovic, M.: Translation the Wiki Way, *Proceedings of the 2006 international symposium on Wikis*, pp. 19–32 (2006).
- [7] Morita, D. and Ishida, T.: Designing Protocols for Collaborative Translation, *In Principles of Practice in Multi-Agent Systems*, pp. 17–32 (2009).
- [8] Hu, C.: Collaborative translation by monolingual users, *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pp. 3105–3108 (2009).
- [9] Hu, C., Bederson, B. B. and Resnik, P.: MonoTrans2: An Asynchronous Human Computation System to Support Monolingual Translation, pp. –.
- [10] Hautasaari, A., Ishimatsu, M., Xia, L. and Ishida, T.: Supporting Multilingual Discussion of Wikipedia Translation with the Language Grid Toolbox, *IEICE Technical Report NLC2009-44(2010-1)*, pp. 67–72 (2010).
- [11] Tanaka, M., Murakami, Y., Lin, D. and Ishida, T.: Language Grid Toolbox:

- Open Source Multi-language Community Site, *4th International Conference on Universal Communication (IUCS2010)*, pp. 105–111 (2010).
- [12] Ishida, T.: Language Grid: An Infrastructure for Intercultural Collaboration, *IEEE/ISPJ Symposium on Applications and the Internet (SAINT-06)*, pp. 96–100 (2006).
- [13] Shigenobu, T.: Evaluation and Usability of Back Translation for Intercultural Communication, *International Conference on Human-Computer Interaction (HCII-07)*, pp. 259–265 (2007).
- [14] Yamashita, N. and Ishida, T.: Effects of machine translation on collaborative work, *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work(CSCW-06)*, pp. 515–524 (2006).
- [15] Winograd, T. and Flores, F.: *Understanding computers and cognition: A new foundation for design* (1986).
- [16] Suchman, L.: Do categories have politics? The language/action perspective reconsidered, *Computer Supported Cooperative Work*, Vol. 2, pp. 177–190 (1994).
- [17] Bullen, C. and Bennett, J.: Learning from User Experience with Groupware, *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, pp. 291–302 (1990).
- [18] Johnson, B. and Weaver, G.: Using a computer-based tool to support collaboration:A field experiment, *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, pp. 343–352 (1986).