

特別研究報告書

共同翻訳のためのプロトコルと支援システム  
の開発

指導教員 石田 亨 教授

京都大学工学部情報学科

平成 20 年 2 月 8 日

森田 大翼

## 共同翻訳のためのプロトコルと支援システムの開発

森田 大翼

### 内容梗概

近年，インターネット等の通信基盤の整備やそれに伴う諸技術の開発が進み，一般ユーザは世界中の情報を入手できるだけでなく，容易に世界中に情報を発信できるようになった．しかし，実際にはそれぞれの人間が入手できる情報は制限され，その主な原因は言語の問題である．

そのため，それらの情報を取得するためには，その文書を翻訳する必要がある．しかし，従来の人手による翻訳では作業は完全に翻訳者の能力に依存し，その翻訳者が扱えない言語には翻訳することができないという問題や，英語が翻訳のハブ言語となる傾向があるという問題がある．

そこで，機械翻訳を用いることで，これらの問題を解決することを考えた．しかし，機械翻訳の精度は100%では無い．そこで，文書の翻訳元言語を理解する人間(以下「原文側」とする)と，翻訳先言語を理解する人間(以下「翻訳文側」とする)が機械翻訳に参加し，これらの参加者が相互作用することで良い翻訳文を作成するという共同翻訳の構想が得られた．ここで，共同翻訳の参加者は相手側の言語を知る必要が無く，2言語間の翻訳は翻訳機のみが行う．よって，その2言語間の翻訳機が得られれば，共同翻訳によりその言語間の文章の翻訳を可能にするため，翻訳が翻訳者の能力に依存する問題を解決する．

共同翻訳における人間同士のインタラクションを考える上で，以下のような問題がある．

#### 1) 人間同士のインタラクションの衝突

両者ともに同一の文章に手を加えるため，一方の作業が他方の作業に支障をきたす可能性がある．その結果，人間にストレスが生じたり，他方の作業に支障をきたす可能性を極度に恐れて作業が抑制されることにより，作業効率を落とすという問題がある．

#### 2) 文意の正確な伝達

翻訳システムが原因で翻訳された文章が原文の意味通りに翻訳されない可能性がある．また，翻訳文を読む人によって文意の捕らえ方に違いが生じる可能性があり，翻訳文の修正によって翻訳文の文意が原文の文意とより異なってしまう可能性がある．

本研究では上記の問題を考慮した上で、品質の高い共同翻訳を効率よく行うためのプロトコルの設計を行う。まず、品質を高くするためには原文の文意が正しく翻訳文に反映されているかを確認できなければならないが、翻訳文側にはそれを判断することができない。翻訳文側は得られた翻訳文から文意を推測して流暢な文に修正を行う。文意の伝達の正しさの確認は、その文をもう一度原文側の言語に翻訳した文と原文を比較することで行う。また、人間同士のインタラクションの衝突を避けるために、プロトコル上で操作権制御を行う。

上記を考慮して、プロトコルの基本的な流れを定義し、基本プロトコルを設計した。しかし、基本プロトコルは作業効率に問題があるため、ヒューリスティクスを用いて指摘された問題を改善することができるプロトコルを基本プロトコルに加えて2つ設計した。

また、これら3種のプロトコルを可能にし、共同翻訳のプロトコルを実現するための支援システムの設計、開発を行った。設計においては、プロトコルから最低限必要な操作及び機能を定義し、インタフェース画面の機能定義や画面設計、画面遷移の設計を行った。また、これらの設計に従ってシステムを実装し、共同翻訳のプロトコルを用いることによって得られる翻訳文の品質の評価、及び3種のプロトコルについての効果の違いの評価を行った。

本研究の貢献は以下の2点である。

- 1) プロトコルの設計による人間同士のインタラクションの衝突の防止  
プロトコルの設計の上で、一方が作業している間は他方の作業を抑制するようにし、人間同士の作業の衝突が起こらないよう留意された。また、プロトコルを実現する支援システムを開発し、ユーザの操作を適切に誘導することにより、ユーザがインタラクションの衝突を意識することなく作業を行うことが可能な環境を構築した。
- 2) 折り返し翻訳の導入による文意の正確な伝達の確認  
原文から得られた翻訳文の修正文を、再度原文の言語に翻訳することによって、翻訳文の意味内容が正しく伝わっているか原文側で確認できるようにした。また、改善されたプロトコルにおいては修正文作成時に折り返し翻訳文を得ることにより、片側のユーザで翻訳文の品質を高めることにより、文意が正確かつ効率的に伝達することを可能にした。

## Designing Protocols and Support Systems for Collaborative Translation

Daisuke Morita

### Abstract

In recent years, the development of Internet and other communication infrastructures and of relevant technologies has made it possible for general users not only to access information about all the world but to transmit the information to the world easily. However, in fact, information available for people is limited mainly owing to the language problem.

Therefore, it is necessary to translate documents to access those information. However, manual translation may cause the problem that it is unable to translate into languages translators don't understand and the problem of English acting as the hub language in translation because of becoming dependent on translators' ability.

Therefore, the purpose of this research is to solve above problems using MT. However, pure MT can't make perfectly translated texts because of the low precision of translation. Then, the idea called "Collaborative Translation" happens that a good translation is done by interaction among someone who understand the source language of some document(called "SL-part" in below) , someone who understand the target language(called "TL-part" in below) and MT. In collaborative translation, a person doesn't have to understand the languages the other person understands. Only MT works an act of translation. In sum, given MT which can translate some two languages, collaborative translation enables to translate interlingually some text and solve a problem of becoming entirely dependent on translators' ability in translation.

There are below problems in humans interaction in collaborative translation.

#### 1) Conflict in humans interaction

Both users process the same text, so one side person's operations are likely to affect another. As a result, the problem happens that this causes people to be stressed and efficiency of work is down by suppression extremely concerning about likeliness to affect the other side's work.

#### 2) Accurate transmission of the meaning of sentences

Owing to MT, there is a chance that a translated text doesn't mean accurately an original text. In addition, each person is likely to grasp the meaning of text

differently, so there is a possibility that a modified translated text become more different meaning from an original text.

With above problems in mind, the purpose of this research is to design protocols to perform the high-quality and effective task of collaborative translation. At first, though it is necessary to make sure that a translated text means accurately an original text for high-quality task, TL-part user can't measure a translated text's accuracy for an original. Therefore, it is confirmed that a translated text means accurately an original by comparing with an original and a text which retranslated fluently modified translation into source language(it's called "back-translation"). In addition, concurrency control is required to avoid a conflict of humans interaction.

In consideration for the above, the basic processing flow of protocol was defined, and the basic protocol keeping to the flow was designed. However, there was a problem of efficiency in basic protocol. Therefore, two improved protocols using heuristics were designed further to basic protocol.

Besides, the support system for three protocols of collaborative translation was designed and developed. Minimum required operations and functions were designed by protocols, and screen design and screen transit design of interface were done. In addition, the support system was implemented according to these design, and evaluations of a quality of a translated text using protocols of collaborative translation and of effects of protocols are performed.

The contributions of this research is as follows.

1) Prevention of conflicts in humans interaction by designing protocols

It was taken care for one person not to have conflicts with the other person by designing protocols one person's work was suppressed during the other person working. Besides, designed support system which did protocols and guided properly users' operation enabled to work without considering interaction conflicts.

2) Confirmation of accurate transmission of sentences meaning by back-translation

Retranslating the text modified a translated text from original into original language enabled SL-part to confirm accurate transmission of the meaning from original to translation. Besides, in improved protocols, getting the back-translation while modifying a text enabled one side user to improve quality of translation. This permitted accurate and effective transmission of the meaning of sentences.

# 共同翻訳のためのプロトコルと支援システムの開発

## 目次

第1章	はじめに	1
第2章	背景	3
2.1	言語グリッド	3
2.2	翻訳エージェント	3
第3章	共同翻訳プロトコル	5
3.1	共同翻訳	5
3.2	基本プロトコル	8
3.3	折り返し翻訳の導入	10
3.4	閾値の上限, 下限の導入	12
第4章	共同翻訳支援システム	15
第5章	実装と評価	19
5.1	システムの実装	19
5.2	プロトコルの評価	19
5.2.1	実験方法	20
5.2.2	評価結果	21
5.2.3	日英以外の言語対に対する実験及び評価	22
5.2.4	考察	23
第6章	おわりに	25
	謝辞	26
	参考文献	27
	付録	A-1
A.1	共同翻訳実験のログ	A-1
A.1.1	基本プロトコルを適用した場合	A-1
A.1.2	折り返し翻訳を用いたプロトコルを適用した場合	A-2
A.1.3	上限, 下限の閾値を導入したプロトコルを適用した場合	A-3

A.2	ソースコード	A-4
A.2.1	Server.java	A-4
A.2.2	ServiceController.java	A-6
A.2.3	UserThread.java	A-14
A.2.4	TranslationThread.java	A-17
A.2.5	SentenceTask.java	A-18
A.2.6	InformationTuple.java	A-19
A.2.7	TranslationClient.java	A-20
A.2.8	Client.java	A-20
A.2.9	Reader.java	A-25
A.2.10	Protocol.java	A-27
A.2.11	JClientFrame.java	A-36
A.2.12	JSettingPerspective.java	A-37
A.2.13	JWaitingPerspective.java	A-43
A.2.14	JMainPerspective.java	A-43
A.2.15	JProcessView.java	A-45
A.2.16	JMessageView.java	A-53
A.2.17	JStateView.java	A-54
A.2.18	JSubMenu.java	A-55
A.2.19	Sentence.java	A-56
A.2.20	ComboBoxPair.java	A-57
A.2.21	OnMouseAction.java	A-57
A.2.22	Pair.java	A-58

## 第1章 はじめに

近年，インターネット等の通信基盤の整備が世界中でなされており，人々はPC等の通信端末を用いてどこにいても世界中の情報を入手できる手段を得た．特にインターネットに関しては，近年様々な技術が開発されており，今や一般ユーザが手軽に情報を入手できるのみならず，容易に情報を発信できるようになった．

しかし，世界中の情報を入手できるということはあくまで信号レベルでの話であり，実際には様々な問題が原因で入手できる情報が制限されている．その最大の問題が言語の問題である．唯一の共通の言語と考えられる英語ですら，それを公用語としない人々には理解することは困難であり，英語を用いて自分の考えを自由に表現することは更に困難である．ましてや世界中には数千の言語があり，その言語の壁を越えることは容易ではない．

この状況において機械翻訳というツールは非常に有用なシステムである．全く理解できない言語の文章も機械翻訳を用いて理解できる言語に変換してしまえば，その文章からの情報を理解し，取得することができる．逆に，作成した文章を様々な言語に翻訳してから発信すれば，世界中に情報を発信することができる．機械翻訳を使えば，インターネット上の言語の問題を根本から解決することができる．しかし，現状では一般に機械翻訳の品質は高くなく，実用には到底使うことができない．

言語の壁を克服するためのコラボレーションの研究の一環として，2002年に日中韓馬のアジア4カ国の大学で母国語を用いてオープンソフトウェアを開発するという異文化コラボレーション実験(ICE2002)[1]が行われた．この実験では，参加者は機械翻訳を利用し，母国語を用いてコミュニケーションを取りながら共同作業をするという制約を課された．この実験によって，機械翻訳を用いたコラボレーションは確かに機械翻訳の品質の低さによる意思伝達の難しさはあったが，母国語でメッセージを書くため，発言の際の心理的バリアを感じることは無いということや，発言者の意見がある程度は汲み取ることができ，ウェアネスの支援になることが観察され，有用なツールとなり得ることが分かった．また，この実験では参加者が翻訳品質が良くなるように文章を修正する様子が観察され，その後文章の修正作業に後述の折り返し翻訳を導入することが提案された．



上記の実験における経験として、機械翻訳研究の新たな評価軸として相互作用性が考えられた。機械翻訳システムが自身の翻訳品質を認識し、翻訳可能かどうかの評価、翻訳できない箇所の特定、翻訳できない箇所の言い換え等を行い、それを翻訳文を作成する人に伝えることができれば、それは良い翻訳文を作成することに役立つと考えられた。そこで、機械翻訳をエージェントのメタファとしてみなし、システムと人間との相互作用を評価軸とする、翻訳エージェントの構想が考えられた。

また、人間と翻訳システムを用いて意味内容が理解できる翻訳文を作成できても、機械翻訳による翻訳文は冗長であったり不自然な言い回しを用いていたりと流暢でない場合が多い。そこで翻訳文の言語を良く理解する人が翻訳システムが生成した翻訳文に手を加えて流暢な文章に仕上げることで、より良い翻訳文の生成が可能となるかもしれない。そこで、互いの言語を理解しない二人の人間が翻訳機を介して共同で文章を翻訳する共同翻訳の構想が考えられた。

共同翻訳を考える上で、以下のような問題がある。

#### 1) 人間同士のインタラクションの衝突

両者ともに同一の文章に手を加えるため、一方の作業が他方の作業に支障をきたす可能性がある。その結果、人間にストレスが生じたり、他方の作業に支障をきたす可能性を極度に恐れて作業が抑制されることにより、作業効率を落とすという問題がある。

#### 2) 文意の正確な伝達

翻訳システムが原因で翻訳された文章が原文の意味通りに翻訳されない可能性がある。また、翻訳文を読む人によって文意の捕らえ方に違いが生じる可能性があり、翻訳文の修正によって翻訳文の文意が原文のそれとより異なってしまう可能性がある。

本研究では、このような問題を解決するよう共同翻訳のプロトコルを適切に設計する。また、共同翻訳の支援システムを実装し、プロトコルの有用性を検証する必要がある。

本稿では第2章で本研究の背景として異文化コラボレーション実験から得られた知見により生まれた構想である言語グリッド [2] と翻訳エージェントについて述べる。第3章では共同翻訳のプロトコルについて基本のプロトコルを設計し、その利点と問題点を明らかにしてプロトコルを洗練していく。第4章では共同翻訳支援システム実装のためのインタフェースデザインを行う。第5章に

において実際に共同翻訳支援システムを用いて翻訳品質及びプロトコルの評価を行う。第6章では、本研究における成果や今後の課題などの考察を行う。

## 第2章 背景

### 2.1 言語グリッド

ICE2002 から得られた知見の一つは、インターネット上の言語資源や言語処理機能のアクセシビリティに関するものであった。機械翻訳システムの機能や品質を評価することは容易ではない。また、機械翻訳システムの購入にかかる契約や費用の問題もあり、機械翻訳システムを利用する以前にもこのような困難さがある。また、利用を開始した後も個々の研究のニーズに合わせて機械翻訳システムをカスタマイズすることも容易ではない。

このように、インターネット上の言語資源や言語処理機能のアクセシビリティには問題があり、異文化コラボレーションをインターネット上の言語資源や言語処理機能で支えるためには、ユーザが自らの活動のための言語サービスを容易に作り出せる仕組みが必要である。このようにして、言語グリッドの研究構想が固まった。

言語グリッド(Language Grid)[2]は、インターネット上の既存のサービスを自由に組み合わせ、新たな言語サービスを生み出して利用したり、ユーザ自らが新たな言語資源や言語処理機能を追加していける枠組みである。言語グリッドには、対訳翻訳や機械翻訳などを組み合わせる「水平型言語グリッド」と、応用に特化し、コミュニティが必要とする言語サービスを生み出す「垂直型言語グリッド」の2種類の機能を持つ。

この言語グリッドの枠組みにより、言語資源や言語処理機能のアクセシビリティが向上し、異文化コラボレーション研究を活性化させることが可能となる。

本研究の共同翻訳や翻訳エージェントの翻訳機能全般はこの言語グリッドのサービスを利用している。

### 2.2 翻訳エージェント

第1章でも述べたとおり、ICE2002 では翻訳文の品質が向上するように入力文を修正する様子(これを自己主導型リペア, Self-initiated repair と呼ぶ)が観察されたが、その修正はたびたび繰り返された。ユーザには入力文のどの部分を

表 2.1: 折り返し翻訳文の利用により翻訳文が改善される例

原文	翻訳文	折り返し翻訳文
病人はまずよく寝ることだ。	It's that a sick person often sleeps first.	それは、病気の人がしばしば最初に眠ることである。
病人は寝ることが一番重要だ。	It's most important for a sick person to sleep.	病気の人が眠ることはとても重要である。

直せば質の高い翻訳結果が得られるかが分からない。そこで、機械翻訳システムには翻訳できない箇所をユーザに伝える、ユーザがその箇所を修正した文章を再度翻訳システムに翻訳可能か問い直す、というような人間とシステムのインタラクションが機械翻訳にも有用であるだろうと考えられた。つまり、従来の機械翻訳システムはメタファとしてトランスペアレントな通信路、評価軸を翻訳品質においてきたが、メタファとしてエージェント、評価軸を相互作用性とする新たな機械翻訳研究の見方ができるのではないかと考えられた。これが翻訳エージェントの構想である。

人間とシステムがインタラクションをするには、機械翻訳システムにメタレベルアーキテクチャが必要である。即ち、システム自身が翻訳結果の翻訳品質を理解し、翻訳品質が低ければ入力文のどの部分が翻訳不可能であるかが分かるための機能が必要である。また、システム自身が翻訳不可能な箇所を翻訳可能にするような言い換えを提示する機能があれば、より人間とシステムのインタラクションが活発化する。

機械翻訳システム自身が翻訳品質を認識するために、入力文と折り返し翻訳文の類似度計算を行い、類似度が大きいほど翻訳品質が高いものであるとシステムは判断する。即ち、同一言語の2文を比較し、2文の類似度を計算する機能 [3] が必要である。ここで折り返し翻訳とは、ある入力文を翻訳したい対象言語の文に翻訳した文を再度入力文の言語に翻訳しなおすことである。折り返し翻訳文の質が良ければその中間に当たる翻訳文の質が良くなる傾向があることはICE2002 やその他の実践で経験的に分かっている [4]。例えば表 2.1 のように、上の例では折り返し翻訳文が原文の意味内容と異なっており、翻訳文も原文の意味内容を示していないが、下の例のように原文を修正すると、その折り返し翻訳文は原文の内容とほぼ同じ意味を表しており、翻訳文も原文の意味内容を示すように改善されている。

入力文のどの部分が翻訳不可能であるかが分かるための機能は、機械翻訳システムは入力文を単純化し、どこまで単純化すれば高い翻訳品質が得られるかを調べることで実現できる [5]。そして、システム自身で翻訳できない箇所を見つけ、ユーザに言い換えを求めることができる。また、ユーザに言い換えを求めるだけでなく、言い換え技術 [6] を用いて翻訳できない箇所を言い換えた文章を作成して再度翻訳品質を評価し、その翻訳品質が言い換え前より向上すればその言い換え候補をユーザに提示することも可能となる。

上記の諸技術については [3][5][6] の研究成果によって実現されている。これらの技術を連結し、プロトタイプとして実装がなされている。

## 第3章 共同翻訳プロトコル

### 3.1 共同翻訳

従来、文章等の翻訳作業は2つの言語を理解する1人、あるいは複数の人間によって行われていた。しかし、その作業は完全に翻訳者の能力に依存し、その翻訳者が扱えない言語には翻訳することができない。また、特に英語を共通に扱える人が多くなるため、人手による文章の翻訳は英語がハブ言語となるという問題が起こってしまう。

共同翻訳では、翻訳する作業を行うものは、2人の人間と翻訳システムである。2人の人間は翻訳する言語のうち片方の言語をそれぞれ理解できれば良く、両方の言語を理解できる必要は無い。2言語間の翻訳は翻訳機のみが行う。よって、その2言語間の翻訳機が得られれば、その言語間の文章の翻訳を可能とする。これは上記にあげたような、翻訳が翻訳者の能力に依存するという問題や、英語がハブ言語となる問題を解決する。

もし、機械翻訳の精度が100%正しいとしたら共同翻訳という作業は全く必要無い。文章を翻訳機にかければそれで作業は終了である。また、精度が0%であれば(例えば、翻訳機を乱文字列生成機とみなせば)おそらくこの枠組みは何の役にも立たないであろう。しかし、実際の翻訳機はその中間の辺りにあり、正しい翻訳をすることもあるが、一部が誤って翻訳したり、完全に誤った翻訳をすることもある。その翻訳機と人間のインタラクションのプロトコルを適切にデザインすることによって、翻訳結果というアウトプットの品質を向上させることを目標にしている。これは翻訳エージェントと同様に機械翻訳の評価軸と

して相互作用性を置くことが機械翻訳研究に有用であることを示すことになる。本稿では、翻訳システムは文章を翻訳する機能に限定しているが、将来は翻訳システムとして2.2章の翻訳エージェントを用いて人間とシステム間でより高い相互作用性を実現し、その評価をしていきたいと考えている。

以下、共同翻訳にプロトコルが無い場合の問題を例を用いて指摘する。例を用いるに当たって、共同翻訳を行う最低限のインタフェースとして以下の仮定を用いる。原文、翻訳文、折り返し翻訳文、翻訳文の折り返し翻訳文を表示するための4つのテキストエリアがあり、原文の中のある一文を修正したら対応する箇所に翻訳による変化が反映され、翻訳文のある一文を修正したら原文以外の対応する箇所に翻訳による変化が反映されるものとする。2人の人間はそれぞれ同時に同じ画面を操作できるものとする。

例1) 翻訳文側の操作に原文側が支障をきたす状況 表3.1に実例を示す。表3.1の上の状況があり、中は翻訳文の修正作業から少し遅れて、翻訳文がその後の翻訳に影響を及ぼす前に原文側の修正が行われたとする。表3.1の下はその原文側の変更により、翻訳文を含め上書きされた結果である。表3.1の中の翻訳文の修正は適切であったにも関わらず、作業の衝突によりそれが反映されない。

例2) 原文側が提供する情報を制御すべき状況 表3.2に実例を示す。表3.2の上の状況があり、中は原文が修正した結果であり、下は折り返し翻訳文が良くないので更に原文側が修正をした結果である。このとき、翻訳文側に表3.2の中の結果を見せることで翻訳文側は何らかの処理を行う可能性があり、無駄な負担がかかる。洗練される前の情報は隠されるべきである。

例3) 折り返し翻訳文が悪くても、翻訳文が悪くない場合に起こる状況 表3.3に実例を示す。表3.3の上の状況があり、原文側にとっては折り返し翻訳文が完全に良いとは確信が持てないため、表3.3の下のように修正を行う。しかし、表3.3の翻訳文はやや不自然ではあるが、十分に原文と同じ意味を汲み取ることができる。ここで翻訳文側が修正すると良い結果が得られるかもしれないが、原文側が折り返し翻訳文が良くないと判断して先に修正した場合、翻訳文側にはどうすることもできない。

上記の例から、共同翻訳はプロトコルを適切に設計して人間の行動を制御する必要がある。

以下、基本のプロトコルを設計し、その有用性と問題点を指摘した上でプロ

表 3.1: 翻訳文側の操作に原文が支障をきたす例

原文	翻訳文	折り返し翻訳文	翻訳文の折り返し翻訳文
まず、自分たちの街を見て、その状況を把握する。	First the town is seen and its situation is grasped.	第一に、町は見られて、その状況は把握される。	First a town is seen, and its situation is grasped.
まず、自分たちの街を見ることによって、その状況を把握する。	At first, we see the town and grasp the situation.(翻訳文側の修正)		
まず、自分たちの街を見ることによって、その状況を把握する。	First its situation is grasped by seeing the town.	最初のその状況は、町を見ることによって把握される。	Its first situation is grasped to see a town.

表 3.2: 提供する情報を制御すべき例

原文	翻訳文	折り返し翻訳文	翻訳文の折り返し翻訳文
ソフトウェアは複雑さを極め、あまりにも多くのものが相互に依存しあっているのです。	Software wins complication, and too much matter is dependent each other mutually.	ソフトウェアは複雑化を勝ち取り、あまりにも多くの物質は相互に依存した互いである。	Software wins complication and too many substances are mutual on which I depend mutually.
ソフトウェアはとても複雑で、とても多くのことが互いに依存しあう。	The software is very complicated and quite many things depend each other each other.	ソフトウェアは非常に複雑で、まったく多くの物は依存するお互いにお互いに。	The software is very complicated and I depend for just many ones, each other Each other.
ソフトウェアはとても複雑で、多くのことが互いに依存する。	The software is very complicated and many things depend each other.	ソフトウェアは非常に複雑で、多くの物は依存するお互いに。	The software is very complicated and I depend for many ones, each other.

表 3.3: 折り返し翻訳文が悪くても、翻訳文は悪くない例

原文	翻訳文	折り返し翻訳文	翻訳文の折り返し翻訳文
この問題は、初期に発見される機会がない。	There are no chances found in an early stage for this problem.	この問題のために初期の段階において発見されたチャンスが全然ない。	There are no chances found in the stage of the early stage for this problem at all.
初期に、この問題が発見されるような機会がない。	There are no chances when this problem is found in an early stage.	この問題が初期の段階において発見される時に、チャンスが全然ない。	When this problem is found in the stage of the early stage, there are no chances at all.

トコルを改善していく。本研究では共同翻訳のターゲットはあるドメインに絞られた文書の翻訳とし、チャットのような対話は扱わない。実際に想定している状況は、人間は翻訳する文書のドメインを良く知っているものとし、また、対話では無いことから、作業の同期性は少々緩和されても良いものとする。

以降、ある翻訳したい文書を持つ翻訳元言語を理解する側の人間を「原文側」、翻訳先言語を理解する側の人間を「翻訳文側」と表現する。

原文を  $s$  , 意味内容が表現される文の accuracy の閾値を  $A$  , 意味内容が流暢に表現される文の fluency の閾値を  $F$  , 修正回数の上限を  $N$  とする . 文章  $s$  の修正を  $m(s)$  , 翻訳を  $t(s)$  , 文章  $t$  との accuracy を  $a(s,t)$  , fluency を  $f(s)$  と表す .

例えば原文  $s$  の修正文の翻訳文は  $t(m(s))$  のように表す .

- 1) 原文側の処理に入る
- 2) 原文の修正文  $m(s)$  の翻訳文  $t(m(s))$  を作成し , 翻訳文側の処理に入る
- 3) もし ,  $f(t(m(s))) < F$  ならば , 翻訳文の修正  $m(t(m(s)))$  の翻訳文  $t(m(t(m(s))))$  を作成して原文側の処理に入り , 5) へ
- 4) もし ,  $F \leq f(t(m(s)))$  ならば ,  $t(t(m(s)))$  を作成して原文側の処理に入る
- 5) 原文側で受け取った文  $s'$  が ,  $A \leq a(s',s)$  ならば , プロトコルを終了し , 成功を返す
- 6) もし ,  $N$  回目の原文側が修正をしていれば , プロトコルを終了し , 失敗を返す
- 7) 2) へ

図 3.1: 共同翻訳の基本プロトコル

### 3.2 基本プロトコル

共同翻訳は , 機械翻訳の問題や CHI の問題を含むため , プロトコルは以下の点に注意して設計されなければならない .

#### 1) 2 人の人間の作業の競合

同一の文章に対して 2 人の人間がそれぞれ独立に作業を行うため , 同一箇所を同時に処理することによって作業の競合が起こる可能性がある . プロトコルは操作権制御を行う必要がある [7] .

#### 2) 機械翻訳による文意の伝達の確認

機械翻訳の場合 , 翻訳文が必ず原文の文意を意味しているとは限らない . 文意が正しく伝達されているか確認する必要がある .

上記の注意点に従った共同翻訳のプロトコルの基本は , 以下のような流れである . 以降 , 文章は 1 文ずつ区切って別々に処理するものとする .

- 1) 原文側が修正した原文を翻訳して翻訳文側に渡す
- 2) 翻訳文側が原文側から受け取った翻訳文から文意を推測して流暢な文に修正し , それを原文側の言語に翻訳し直して , 原文側に渡す
- 3) 原文側が翻訳文側から受け取った折り返し翻訳文が原文の意味内容を表しているかを評価し , 正しく翻訳されているかどうかを確認する

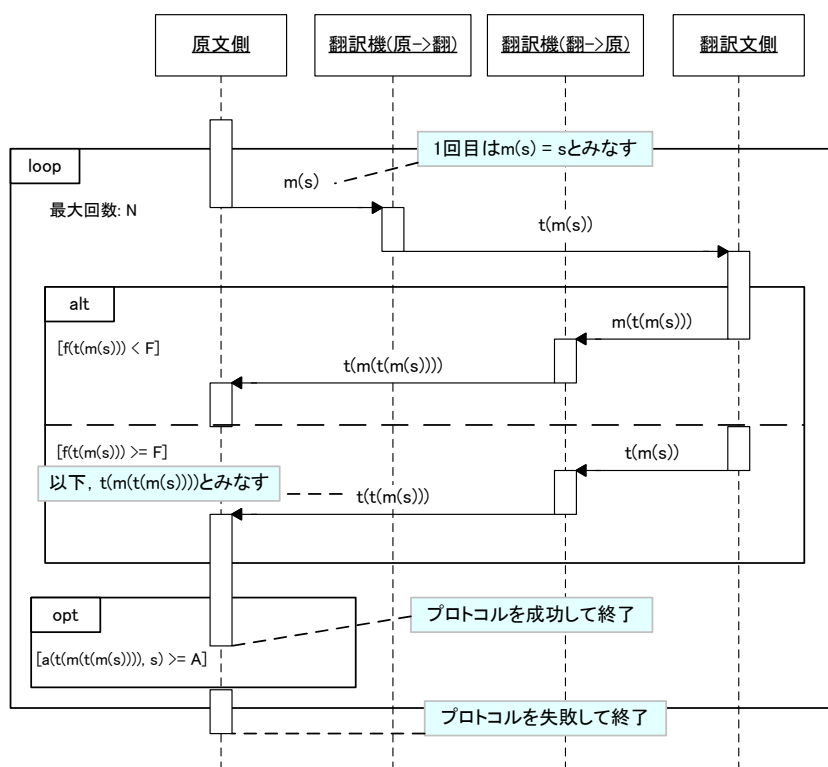


図 3.2: 基本プロトコルのシーケンス図

上記において、翻訳文側が持つ文章の評価基準は流暢さ (Fluency) である。翻訳文側は翻訳文を見て原文と同じ意味を持っているかどうかを判断することはできないため、翻訳文側にできることは文章を自然な文に修正することのみである。また、人間によって修正された文章は必ず fluent であるとする。

代わりに、原文側が意味内容の正しさ (Accuracy) を評価する。そのため、翻訳文側に修正された文を再度原文の言語に翻訳し直し、その文と原文を比較することによって翻訳文側に意味内容が正しく伝わったかを確認する。

上記の流れをプロトコルにしたものが基本となる。図 3.1 にプロトコルを示す。また、図 3.2 にこのプロトコルのシーケンス図を表す。

表 3.4: 表 3.1 の例に基本プロトコルを適用した場合

原文	翻訳文	折り返し翻訳文
まず、自分たちの街を見て、その状況を把握する。	First the town is seen and its situation is grasped.	
まず、自分たちの街を見て、その状況を把握する。	At first, we see the town and grasp the situation.	最初は、私達は町を見て、状況を把握する。



表 3.4 に上記表 3.1 の例に基本プロトコルを適用した場合を示す．表 3.4 の上の状況は，折り返し翻訳が見えないという点以外は表 3.1 と同様であるが，ここで操作権は明確に翻訳文側に移るため，翻訳文側は下のように修正し，その文の翻訳により折り返し翻訳文が得られ，確認して共同翻訳を成功させることができる．このように，このプロトコルによって人間同士の作業の衝突を避けることができる．

しかし，このプロトコルでは良くない原文の修正に対しても同様に翻訳文側に処理が移り，無駄なインタラクションが生じるという問題がある．表 3.2 に示されるような問題がある．これを改善するプロトコルを 3.3 章に示す．

### 3.3 折り返し翻訳の導入

原文を  $s$ ，意味内容が表現される文の accuracy の閾値を  $A$ ，意味内容が流暢に表現される文の fluency の閾値を  $F$ ，修正回数の上限を  $N$  とする．文章  $s$  の修正を  $m(s)$ ，翻訳を  $t(s)$ ，文章  $t$  との accuracy を  $a(s,t)$ ，fluency を  $f(s)$  と表す．

- 1) 原文側の処理に入る
- 2) もし，修正原文  $m(s)$  を作成して， $a(t(m(s)), s) < A$  ならば，
  - a. もし，原文側の修正が  $N$  回目なら，プロトコルを終了し失敗を返す
  - b. 2) へ
- 3) 翻訳文側の処理に入る
- 4) もし， $F \leq f(t(m(s)))$  ならば，プロトコルを終了し成功を返す
- 5) もし，修正翻訳文  $m(t(m(s)))$  を作成して， $a(t(m(t(m(s))))), m(t(m(s))) < A$  ならば，
  - a. もし，翻訳文側の修正が  $N$  回目ならば，プロトコルを終了し，失敗を返す
  - b. 5) へ，
- 6) 原文側の処理に入る
- 7) もし， $a(m(t(m(s))), s) \geq A$  ならばプロトコルを終了し，成功を返す
- 8) もし，翻訳文側の修正が  $N$  回目ならば，プロトコルを終了し，失敗を返す
- 9) 翻訳文側の処理に入り，5) へ，

図 3.3: 共同翻訳の折り返し翻訳を用いたプロトコル

本章で述べるプロトコルは 3.2 章のプロトコルにおける原文の修正文の送信前，翻訳文の修正文を送信する前に折り返し翻訳文を取得し，その折り返し翻

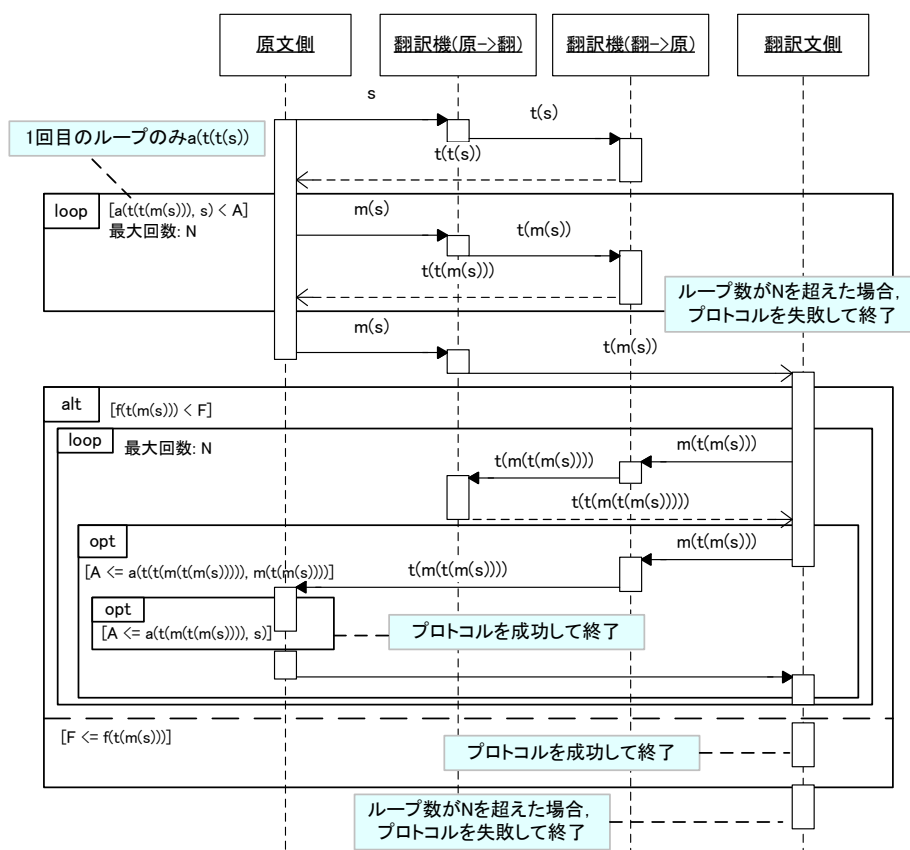


図 3.4: 折り返し翻訳を用いたプロトコルのシーケンス図

訳文の accuracy を良くするように修正して、送信するその間の訳文の品質を前もって高めるといふプロトコルである。ここでも 3.2 章と同様に折り返し訳文の意味内容が原文のそれを良く表しているなら、その訳文は正しい訳文であるというヒューリスティクスを用いている。このプロトコルを図 3.3 に示す。また、図 3.4 にこのプロトコルのシーケンス図を表す。

表 3.5 に表 3.2 の例にプロトコルを適用した例を見せる。表 3.5 の 1 行目から

表 3.5: 表 3.2 の例に折り返し翻訳を用いたプロトコルを適用した場合

原文	訳文	折り返し訳文
ソフトウェアは複雑さを極め、あまりにも多くのものが相互に依存しあっているのです。		ソフトウェアは複雑化を勝ち取り、あまりにも多くの物質は相互に依存した互いである。
ソフトウェアはとても複雑で、とても多くのことが互いに依存しあう。		ソフトウェアは非常に複雑で、まったく多くの物は依存するお互いにお互いに。
ソフトウェアはとても複雑で、多くのことが互いに依存する。	The software is very complicated and many things depend each other.	ソフトウェアは非常に複雑で、多くの物は依存するお互いに。

3行目までのように，原文側では折り返し翻訳文が原文の意味内容を十分に表すようになるまで修正をし，その間翻訳文側はこの文の処理に関与することが無い．原文側が修正を終了したときに翻訳文側ははじめて表 3.5 の 3 行目に示される翻訳文を見る．これにより，提供される情報には無駄が少なく，即ち無駄なインタラクションが減ることになり，表 3.2 に示される問題は解決される．

しかし，図 3.3 のプロトコルでは原文側で修正文を作成し，その折り返し翻訳文の品質が十分良くなる限り，翻訳文側に操作権は移動せず，原文側は修正を繰り返さなければならない．これは 1 文あたりの原文側の処理の負荷を大きくし，また，表 3.3 に示されるように，折り返し翻訳文の品質が悪い場合でも，翻訳文の品質は良い場合があり，本プロトコルではその翻訳文を翻訳文側に処理させることができないという問題がある．この問題を解決するために，3.4 章のプロトコルでは新たなヒューリスティクスを導入し，修正文の誤った棄却を可能な限り防ぐよう改善する．

### 3.4 閾値の上限，下限の導入

3.2 章，3.3 章のプロトコルでは，文の accuracy，あるいは fluency の値が閾値に対して大きい値であれば，十分意味内容を表している，あるいは十分流暢であるとして，修正をする余地が無いとみなしていた．即ち，修正すべき文の上限の閾値を表していたと考えられる．本章では，この上限の閾値に対して，下限の閾値を設ける．そこで，図 3.5 のプロトコルに示すように図 3.1，3.3 の A，F は図 3.5 において UA，UF と再定義し，新たに LA，LF という定数を定義する．また，図 3.6 にこのプロトコルのシーケンス図を表す．

本章のプロトコルでは，ある文に対する折り返し翻訳文から原文の意味内容が推測できないのであれば，その翻訳文の意味内容は原文のそれをほとんど表せていないというヒューリスティクスと，折り返し翻訳文から原文の意味内容がある程度推測できるのであれば，その翻訳文の意味内容は原文の意味内容を推測できる程度表しているというヒューリスティクスを用いている．fluency も同様に内容が推測できないほど翻訳文が流暢でないなら，それを修正のしようがないと考えられる．本章のプロトコルでは修正する価値のある文の accuracy と fluency の値は，その閾値の上限と下限の間にある文であると考えられることができる．

表 3.6 に表 3.3 の例にプロトコルを適用した例を示す．表 3.6 上の折り返し文

原文を  $s$  , 意味内容が推測できる文の accuracy の閾値を  $LA$  , 意味内容が表現される文の accuracy の閾値を  $UA$  , 意味内容が推測できる文の fluency の閾値を  $LF$  , 意味内容が流暢に表現される文の fluency の閾値を  $UF$  , 修正回数の上限を  $N$  とする . 文章  $s$  の修正を  $m(s)$  , 翻訳を  $t(s)$  , 文章  $t$  との accuracy を  $a(s,t)$  , fluency を  $f(s)$  と表す .

- 1) 原文側の処理に入る
- 2) もし , 原文側の修正を  $N$  回行ったなら , プロトコルを終了し失敗を返す
- 3) もし , 修正原文  $m(s)$  を作成して ,  $a(t(t(m(s))), s) < LA$  ならば , 2) へ
- 4) もし ,  $LA \leq a(t(t(m(s))), s) < UA$  ならば ,
  - a. 翻訳文側の処理に入る
  - b. もし ,  $f(t(m(s))) < LF$  または  $UF \leq f(t(m(s)))$  ならば , 1) へ
  - c. もし , 修正翻訳文  $m(t(m(s)))$  を作成して ,  $a(t(t(m(t(m(s))))), m(t(m(s)))) < LA$  ならば ,
    - i. もし , 翻訳文側の修正が  $N$  回目なら , 1) へ
    - ii. c. へ ,
  - d. 原文側の処理に入る
  - e. もし ,  $UA \leq a(t(m(t(m(s))))), s$  ならば , プロトコルを終了し , 成功を返す
  - f. 2) へ
- 5) 翻訳文側の処理に入る
- 6) もし ,  $f(t(m(s))) < LF$  ならば , プロトコルを終了し , 失敗を返す
- 7) もし ,  $UF \leq f(t(m(s)))$  ならば , プロトコルを終了し , 成功を返す
- 8) もし , 修正翻訳文  $m(t(m(s)))$  を作成して ,  $a(t(t(m(t(m(s))))), m(t(m(s)))) < LA$  ならば ,
  - a. もし , 翻訳文側の修正が  $N$  回目なら , プロトコルを終了し , 失敗を返す
  - b. 8) へ ,
- 9) 原文側の処理に入る
- 10) もし ,  $a(t(m(t(m(s))))), s \geq UA$  ならばプロトコルを終了し , 成功を返す
- 11) もし , 翻訳文側の修正が  $N$  回目ならば , プロトコルを終了し , 失敗を返す
- 12) 翻訳文側の処理に入り , 8) へ ,

図 3.5: 共同翻訳の折り返し翻訳と上限下限の閾値を用いたプロトコル

は良くは無いが , 意味内容を推測できないほどでは無いため , 翻訳文側に修正をさせることができる . 表 3.6 の下のように , 翻訳文側が修正を行った場合 , その折り返し翻訳文は推測できる程度は意味内容を表しているのので , 原文側に評価を要求できる . 原文側は折り返し翻訳文が原文の意味内容を表しているのので ,

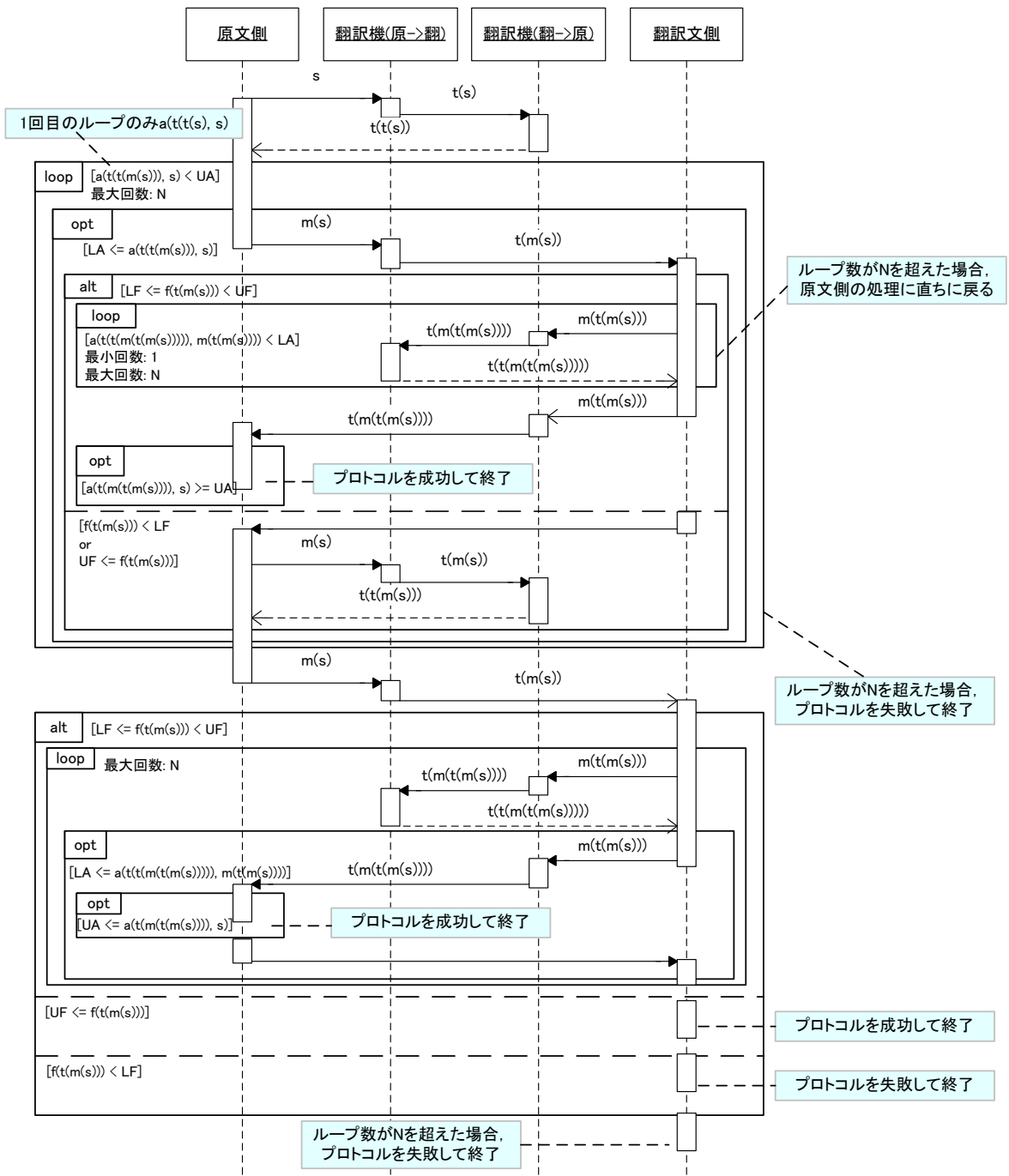


図 3.6: 折り返し翻訳と上限下限の閾値を用いたプロトコルのシーケンス図

成功でプロトコルを終了できる。即ち、本章のプロトコルを用いることにより、3.3章のプロトコルで問題であった翻訳文の品質が良いが、折り返し翻訳文の品質が悪いため棄却されるという問題を解決することができる。

表 3.6: 表 3.3 の例に上限下限の閾値を導入したプロトコルを適用した場合

原文	翻訳文	折り返し翻訳文	翻訳文の折り返し翻訳文
この問題は、初期に発見される機会がない。	There are no chances found in an early stage for this problem.	この問題のために初期の段階において発見されたチャンスが全然ない。	
初期に、この問題が発見されるような機会がない。	There are no chances that this problem is found in an early stage.	この問題が初期の段階において発見されるという可能性が全然ない。	There are no possibilities that this problem is found in the stage of the early stage at all.

## 第4章 共同翻訳支援システム

システムは文書を複数の文に分割して、それぞれの文について独立にプロトコルを適用し、並行して翻訳作業を行うものとする。即ち、2人のユーザは互いの作業を待つことなく、処理可能な文から次々に可能な作業を行うものとする。例えば、原文側が文を修正し、翻訳文側に送信したとしても、原文側はその文の折り返し翻訳を待つことなく、次の文の修正や確認作業に入る。翻訳文側がその文についての修正が終わったときに、原文側はその折り返し翻訳文の確認作業を行う。これによって、システムの作業効率の向上を測ることができる。

以下、第3章にて定義したプロトコルからシステムのインタフェースに必要な機能を定義する。また、そのプロトコルを満たすために各画面がどのように遷移するかを設計する。

第3章に示された3つのプロトコルにおいて、各ユーザの状態は修正状態、確認状態、待機状態、終了状態がある。各プロトコルに依存してユーザの状態は変化をする。システムのユーザインタフェース画面は、その状態の変化に依存して画面遷移を行う。

以下に、各状態についての説明と必要なインタフェースを述べる。

### 1) 修正状態

原文側なら原文を、翻訳文側なら原文の修正に対する翻訳文から修正文を作成する。よって、原文や翻訳文の表示機能と、その近くに文の修正を記述するためのテキストエリアが必要である。また、修正の完了を決定するための機能(ボタン等)が必要である。

### 2) 確認状態

折り返し翻訳文の **accuracy** の確認 原文側なら原文を、翻訳文側なら翻訳文を見ながらその文の折り返し翻訳文の **accuracy** の評価を行う。その

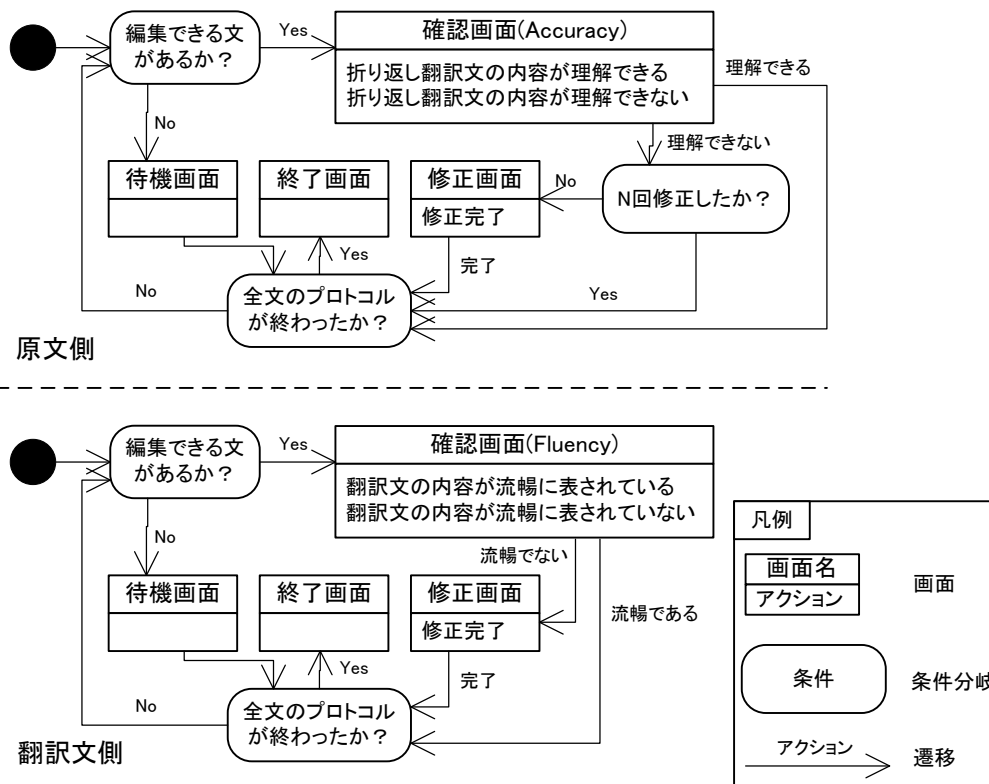


図 4.1: 基本プロトコルの画面遷移図

ため、原文や翻訳文の表示機能と、その文に対する折り返し翻訳文を表示する機能が必要である。また、その accuracy の評価をするため選択肢を表示し、決定させる機能が必要である。

**翻訳文の fluency の確認** 翻訳文側のみにある状態であり、翻訳文を見ながら fluency の評価を行う。そのため、翻訳文の表示機能が必要である。また、その fluency の評価をするための選択肢を表示し、決定させる機能が必要である。

### 3) 待機状態

文書中の各文が相手側の処理の状態にあるか、プロトコルを終了しているかのいずれかである状態。作業可能な状態になる文が表れるまで「待機中」などシステムが待機状態であることを示すメッセージを表示する機能が必要である。

### 4) 終了状態

文書中の全ての文がプロトコルを終了している状態。システムが全ての文を

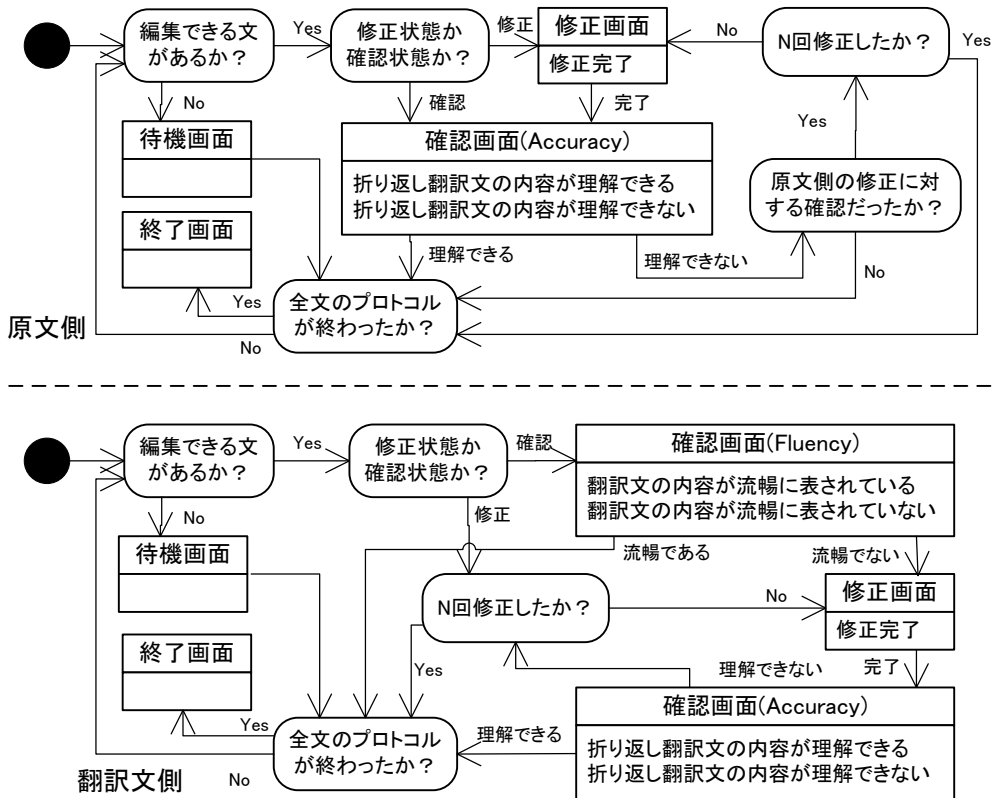


図 4.2: 折り返し翻訳を用いたプロトコルの画面遷移図

処理し終えたことを示すメッセージを表示する機能が必要である。

これらの状態における機能を定義した上で、図 4.1、4.2、4.3 に各プロトコルの画面遷移図を示す。

これらの画面遷移図は、図 4.1 に示される凡例に従っている。画面は長方形を線で分割したもので表し、上に画面の名前、下にその画面上で行うことができるアクションの集合で表される。その画面上で、アクションが実行されるとそのアクションに従って画面が遷移する。条件分岐は角丸四角で表し、その条件文は中央に表記されている。画面上で行うアクションの他にその時点におけるシステムの状態や各文の処理経過の状態によって画面遷移先が異なる場合に用いられている。遷移は矢印で表現される。矢印に付いているラベルは、遷移元におけるアクションを表し、そのアクションに従って遷移する。ラベルが無い遷移は任意のタイミングで遷移できる。

基本プロトコルについて、この画面遷移図の正しさを証明する。図 3.2 より、原文側が活性化される時はプロトコル開始時か翻訳文側の処理を終えたとき



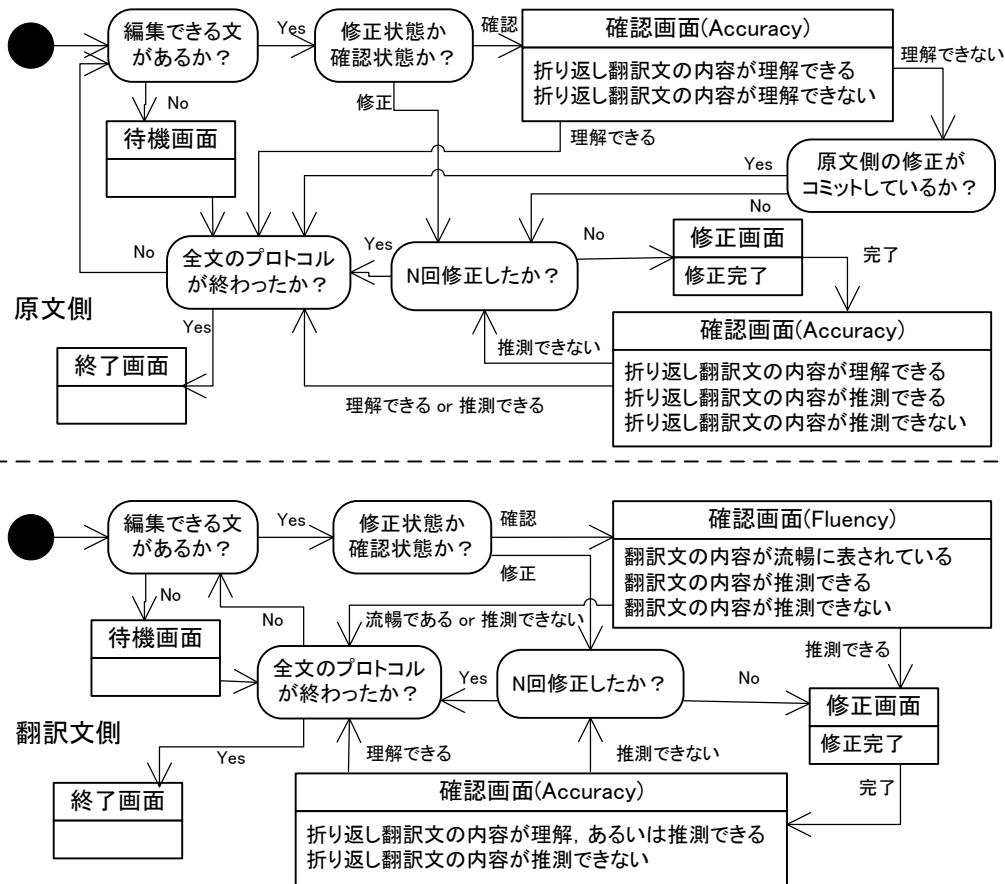


図 4.3: 折り返し翻訳と上限下限の閾値を用いたプロトコルの画面遷移図

である。前者の場合，原文を処理することなく翻訳文側に翻訳して送信するため，原文側が何らかの処理を行うことは無い。後者の場合，原文側は確認状態（原文と，翻訳文側によって修正された翻訳文からの折り返し翻訳文の accuracy の評価を行う状態）から始まる。そのため，処理可能な文が見つかった場合は必ず確認状態に遷移する。図 3.2 のシーケンス図から以下のことが読み取れる。もし，折り返し翻訳文の意味内容が理解できればその文の処理を終了し，次の文の処理に移る。そうでないなら，図 3.2 のループの最大回数は  $N$  であるので，修正が  $N$  回なされていれば文の処理を終了し，次の文の処理に移る。そうでないなら，修正状態に移動し，修正が完了したら翻訳文側に翻訳文を送信して文の処理を終了し，次の文の処理に移る。図 4.1 は上記の処理過程を表現できている。翻訳文側についても同様にして，図 3.2 のシーケンス図と図 4.1 の画面遷移図が対応していることが分かる。

## 第5章 実装と評価

### 5.1 システムの実装

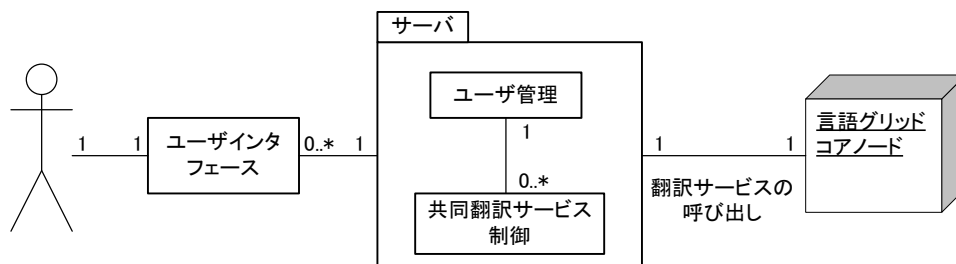


図 5.1: システム構成

共同翻訳支援システムの構成を図 5.1 に示す。本ツールはユーザインタフェースを持つクライアントと、ユーザを管理し、サービスを制御するサーバに分類される。クライアントはサーバにコネクション要求を出し、ユーザ管理部と接続を確立する。ユーザ管理部はコネクションを確立すると同時にユーザ情報を受け取り、接続を受け入れたユーザのマッチングを行う。ここでユーザのマッチングとは、ユーザが原文側か、翻訳文側かという情報や、使用言語、対象言語、使用プロトコルに関する情報から、サービスを開始することができるユーザのペアを見つけ出す。ペアが見つけ出されたら、共同翻訳サービス制御部を新たに生成し、2 ユーザ間のサーバを介した通信を開始する。

本ツールは、クライアントシステム上で修正された文章や対応する翻訳文や折り返し翻訳文はサーバで管理され、ユーザインタフェースに表示される文書情報は全てサーバを介して取得される。本ツールの翻訳モジュールは 2.1 章に説明した言語グリッドによって提供されている Web サービスを用いている。

### 5.2 プロトコルの評価

第 3 章に示した 3 種のプロトコルの評価基準を以下に示す。

**共同翻訳成功率** 文書にプロトコルを適用した場合、文書に含まれる文がどのくらいの割合で成功でプロトコルを終了し、結果を出力するかを示す。これが良いほど処理可能な文書が増え、作業効率が良いことを示す。

**翻訳文の原文表現度** プロトコルを成功した翻訳文の中で品質が高い翻訳文の割合を示す。これが良いほど、成功でプロトコルを終了して得られた翻訳文書の品質の信頼度が高いものとなる。

上記の評価基準により評価するためのデータ測定のため、5.2.1章に示す実験方法で実験を行い、評価を行った。また、5.2.3章のように日英以外の言語対に対し、予備実験の形で実験及び評価を行い、様々な言語で共同翻訳が有効に作用するかの評価を行った。また、それを受けて考察を5.2.4章に示す。

### 5.2.1 実験方法

本実験では用いる文書の言語は日本語とし、共同翻訳システムを用いて英語に翻訳することをタスクとした。また、共同翻訳の適用の場を想定し、使用する文書はドメインを特定したものであり、被験者に理解できるドメインとした。

上記の評価基準で評価するために、各文書に3つのプロトコルを適用する必要がある。しかし、文書の内容の学習を防ぐため、被験者は実験の際に2度同じ文書进行处理してはならない。この点や実験におけるユーザにかかる負担、得られる評価データの質と量を考慮し、以下のように実験方法を設定した。

- 各文書は10文程度
- 被験者は日本語を扱う人3人と英語を扱う人3人で構成される
- 各被験者はそれぞれペアを構成するが、それぞれ別室にて実験を行う
- 各被験者は3つのプロトコルを基本プロトコルから順に行う
- 各被験者は3つのプロトコルにおいて、全て異なる文書を使う

また、各被験者には実験に先立ち、共同翻訳を行う上で必要な最低限の知識として、以下の事前知識を共有させた。

- 翻訳する文書のドメイン
- 折り返し翻訳文の意味とその利用方法
- 3.2章冒頭に示したプロトコルの基本的な流れ

また、実験によって得られた各翻訳文は3人の評価者により評価された。人

翻訳結果において原文の意味をどの程度理解できるか

5 完全に理解できる (All)

4 大部分を理解できる (Most)

3 大体理解できる (Much)

2 ほんの少ししか理解できない (Little)

1 全く理解できない (None)

図 5.2: 翻訳文の品質の評価基準

手による文章の修正は必ず流暢であるとしているため、評価は Accuracy についてのみ行い、プロトコルによって翻訳文に正確に意味が伝達したかを評価した。その評価尺度として図 5.2 のように基準を設定した [8]。

この実験を行うにあたって、結果は以下の様に予想された。なお、以降 3.2 章の基本プロトコルをプロトコル 1、3.3 章の折り返し翻訳を用いたプロトコルをプロトコル 2、3.4 章の上限、下限の閾値を用いたプロトコルをプロトコル 3 と表記する。

共同翻訳が成功する場合、その平均評価値は 4 を超える。共同翻訳では、文意の正しい伝達を原文側が確認を行っている。そのため、共同翻訳が成功する場合、評価は基本的に 5 となると想定され、3 人の評価者の平均をとれば少なくとも一人は 5 と評価し、平均評価値は 4 を超えると考えられる。共同翻訳による改善度はプロトコル 3 が最も大きく、プロトコル 1 が最も小さい。プロトコル 1 に対して、3.3、3.4 章に示したヒューリスティクスを導入し、プロトコルを改善しているため、このように想定される。また、上記の想定も合わせると、共同翻訳成功率もプロトコル 3 が最も大きく、プロトコル 1 が最も小さいと想定される。

### 5.2.2 評価結果

表 5.1 に各プロトコルの共同翻訳成功率を示す。予想とは異なり、表 5.1 に示したように、プロトコルが複雑化していくにつれて共同翻訳成功率が下がっていく様子が観察された。

また、各プロトコルにおいて共同翻訳が成功した場合の文の評価値の分布と機械翻訳のみで得られる文の評価値の分布を図 5.3 に示す。ここで、各文の評価値は 3 人の評価者によって図 5.2 に示される基準によって評価された値の平均値を用いている。図 5.3 に見られるように、機械翻訳のみを用いた場合には「大部分を理解できる」という評価基準である値 4 を超える評価を受けたものは全体の 7% に過ぎなかった。しかし、共同翻訳を用いて成功でプロトコルを終了できた場合、4 を超える評価を受けたものはプロトコル 1 で 37%、プロトコル

表 5.1: 共同翻訳成功率

プロトコル 1	プロトコル 2	プロトコル 3
90.0%	73.3%	66.7%

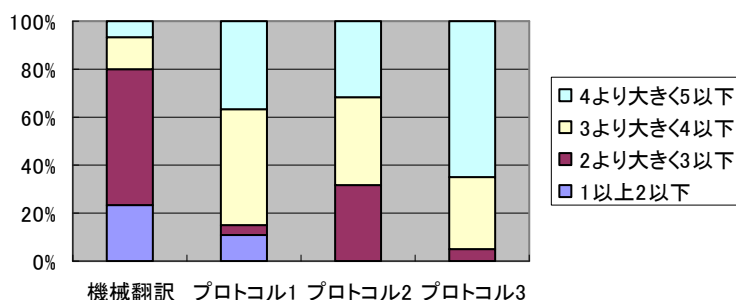


図 5.3: 共同翻訳を成功した文の評価の分布

ル2で32% プロトコル3では65%にまで上った。

また、本実験により得られた修正のログからある文に対する修正処理においてプロトコルの改善により修正作業がより効率化されたケースがいくつか観察された。特に典型的に見られた処理過程のログを付録 A.1 に示す。また、それを解析したデータを表 5.2 に示す。プロトコル1では、原文側の修正による翻訳文毎に評価されるため、インタラクションが非常に多く無駄な処理が増えており、多くの時間がかかっている。プロトコル2では、原文側の修正回数はプロトコル1とさほど差は無いが、折り返し翻訳文が良くなるまで連続して修正を続けるため、比較的効率良く処理がなされている。翻訳文側では、得られた翻訳文が洗練されたものであるため、少ない修正回数で良い修正を作成できている。プロトコル3では原文側は折り返し翻訳文が推測できる程度になった時点で翻訳文側に処理を渡している。翻訳文側は意味を汲み取り、適切な修正を行うことができたため、全体として非常に効率良く作業がなされている。

### 5.2.3 日英以外の言語対に対する実験及び評価

実験に用いる文書は 5.2.1 章と同じ日本語の文書とし、対象言語を中国語、韓国語、フランス語とし、5.2.1 章と同様の方法、評価尺度で実験及び評価がなさ

表 5.2: 付録 A.1 のログの解析データ

	プロトコル1	プロトコル2	プロトコル3
原文側修正回数	5	4	1
翻訳文側修正回数	6	3	2
総インタラクション回数	11	2	2
原文側処理時間	8分22秒	4分10秒	1分39秒
翻訳文側処理時間	3分48秒	1分41秒	2分23秒

表 5.3: 中韓仏の言語に各プロトコルを適用した時の共同翻訳成功率

	プロトコル 1	プロトコル 2	プロトコル 3
日中	90%	100%	80%
日韓	100%	80%	100%
日仏	100%	80%	70%

れた。但し、本実験は各言語対のペアは1組、評価者は1人で行われた。

本実験において結果は以下の様に予想された。

共同翻訳が成功する場合、その評価値は5となる。今回、評価者は1人のため、共同翻訳が成功した場合、評価値は5となると想定された。

韓国語の場合、ほとんどの文が成功する。日韓の翻訳エンジンの翻訳精度はかなり高い。そのため、共同翻訳はほとんどの場合成功すると考えられた。

表 5.3 に各言語にプロトコルを適用した場合の共同翻訳成功率を示す。表 5.3 には日英の実験のように顕著には表れなかったが、概ね日英と同様にプロトコルが複雑になるにつれて共同翻訳成功率が小さくなる傾向が見られた。

また、図 5.4 に日中、日韓、日仏、日英の言語対について共同翻訳を成功でプロトコルを終了した場合の評価値の分布を示す。評価値1~5で区別しているが、英語の場合は4より大きく5以下の平均評価値のものを5、3より大きく4以下の平均評価値のものを4、というように換算している。図 5.4 の評価において、プロトコルの区別はされていない。図 5.4 に示すように、日韓においては80%以上、日中においては70%というように高い割合で評価値が5となっている。一方、日仏は40%程度であり、日英とほぼ同様の傾向を示している。

#### 5.2.4 考察

本実験では、実験前に立てられた予想に反して、評価値が5とならなくてもプロトコルを成功で終了している文が多数見られた。実験のログを分析した結

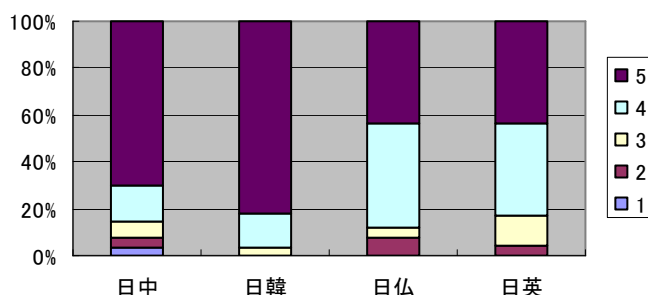


図 5.4: 共同翻訳によって得られた文の翻訳品質

果、それらの大部分は以下のいずれかに当てはまった。

- 作業者への心理的負担による判断基準の歪み
- 翻訳機への対応による過度の文章の簡略化による意味の損失
- 折り返し翻訳文が有効に働かない場合

作業自体が非常に単調であり、また、文章の修正で折り返し翻訳文が一向に改善されない場合にストレスが生じ、ユーザは早くその作業を終了させることを望んだ。また、折り返し翻訳文はこれ以上改善しないと感じた。そのような場合に、折り返し翻訳文が完全でなくてもある程度理解できれば「理解できる」を選択してしまうことにより、不十分な訳文が生成される。また、折り返し翻訳文を品質が高いものにするために、文章を過度に簡略化する動作が見られた。この動作も、文章の修正に対する心理的な負担から逃れるための行動と考えられる。その場合の折り返し翻訳文は原文を完全に意味してはいないが、上記と同様に「理解できる」を選択し、不十分な訳文が生成される。また、翻訳文の意味を誤解して修正された翻訳文に対する折り返し翻訳文が原文と類似しているために、原文側が「理解できる」と判断し、翻訳文が原文の意味と異なる場合も少数ではあるが見られた。

上記2つの問題への対処法としては、2.2章に示した翻訳エージェントの導入により、作業者の負担を軽減することや折り返し翻訳文の評価の主観を排除することによって、人為的な判断基準の歪みの問題を解消できると考えられる。また、原文の過度な修正には修正文と原文の accuracy を確認するプロトコルを追加することで解決できると考えられる。また、それに伴って適切なインタラクション設計を行い、インタフェースを洗練することにより、機能が増えた場合でもそれをユーザが自然に使用することを可能とすることが考えられる。これらは今後の課題とする。

また、プロトコル1に対してプロトコル2の共同翻訳による改善度は高くなかった。プロトコル1やプロトコル3の実験で、原文側ユーザが翻訳文側ユーザの修正に対する折り返し翻訳文を見て、修正のヒントを得ている様子が観察されたが、プロトコル2では修正文作成時にユーザ間の相互作用が行われなかったため、これが原因で品質の改善度が高くならなかったと推察できる。

上記の考察により、実験では人為的な要因のために共同翻訳のプロトコルは完全には有効に働かなかったが、システムの改善で人間にかかる負担を減らし、また、プロトコルの改善で更に適切なユーザの誘導を行うことによって、共同

翻訳が成功した場合に品質が完全に良い翻訳文を生成できるようになる可能性が示された。一方で、プロトコル3における品質改善度は他に比べて非常に高く、ユーザ間の適切なインタラクションデザインは翻訳精度の向上に有効であることが示された。

## 第6章 おわりに

本研究では、精度が完全でない機械翻訳に、文書の翻訳元言語に堪能な人間と、翻訳先言語に堪能な人間とが関わって相互作用を行うことによって精度の高い翻訳を行うという共同翻訳を対象とした。

共同翻訳には機械翻訳に生じる翻訳誤りによって文意が正確に伝達されることが保証されないという問題や、コンピュータを介したコラボレーションにおける人間同士のインタラクションの衝突の問題がある。本研究では、このような問題点を明らかにし、プロトコルを設計するという方針で解決を試みた。プロトコルは以下の様に設計された。

### 1) 人間同士のインタラクションの衝突の防止

プロトコルの設計の上で、一方が作業している間は他方の作業を抑制するようにし、人間同士の作業の衝突が起こらないよう留意された。

### 2) 折り返し翻訳の導入による文意の正確な伝達の確認

原文から得られた翻訳文の修正文を、再度原文の言語に翻訳し、その折り返し翻訳文の意味内容が原文の意味内容を表しているかどうかを確認することによって、翻訳文が原文の意味内容を正しく表しているかどうかを原文側で確認させるように設計された。また、基本プロトコル以外のプロトコルにおいては修正文作成時に折り返し翻訳文を得ることによって、片側のユーザで翻訳文の品質を高め、文意が正確かつ効率的に伝達することを可能にした。

上記の設計方針に従い、3種のプロトコルを設計した。この3種のプロトコルを実現するための支援システムを設計し、ユーザがインタラクションの衝突を意識することなく、効率的に作業を行うことを可能にした。

また、設計されたシステムを実装し、3種のプロトコルの評価をするために実験を行った。実験で得られたデータから、各プロトコルの適用により生成された翻訳文の一定の改善が確認されたが、完全には改善されない場合も観察された。ユーザの作業負荷等の原因が考えられ、今後更に改善が必要である。



また、文章を修正する上で、翻訳文側の修正による反映を原文側が修正のヒントとして扱っている状況も観察され、プロトコルに相互作用性を高めることが有効であることが確認された。一方、共同翻訳の処理過程におけるログの解析により、プロトコルの改善によって作業が効率化されているような文の処理もいくつか確認された。

しかし、本稿では未解決の問題もある。共同翻訳というコンピュータを介したコラボレーションを考える上で、プロトコルによって作業中に共同作業者のアウェアネスを得ることはできない。また、適切な情報共有も困難であり、上述のようにユーザへの文章の修正作業の負担も大きい。

今後の課題としては、2.2章で述べた翻訳エージェントを共同翻訳の翻訳システムとして導入し、更に人間とシステム間の相互作用性を高めることによって共同翻訳の精度の向上や高い効率化、及びユーザの負担の軽減などを測ることを考えている。現在の共同翻訳システムでは accuracy, fluency の評価は人の主観によるものであったが、それをシステムにより自動化する、また、文章の中で翻訳システム自身が翻訳できない箇所を指定し、ユーザに修正のターゲットを絞らせるなどが考えられる。

また、それに関連して、人間とシステムの接点におけるインタラクションの観点 (CHI) や、コンピュータ支援による共同作業という観点 (CSCW) から共同翻訳システムのインタフェースをユーザ中心の設計となるように洗練し、人間の自然で豊かなインタラクションを引き出すことで相互作用性を高め、共同翻訳の品質を向上させることを考えている。人間が適切にメンタルモデルを形成し、対話的にシステムを使用することを可能にするデザイン設計や、情報共有やアウェアネスの支援のための設計を行うことが考えられる。

## 謝辞

本研究を行うにあたり、熱心なご指導、ご助言を賜りました石田亨教授に厚く御礼申し上げます。また、様々な言語処理技術を提供していただき、サポートしていただきました独立行政法人情報通信研究機構の内元清貴氏に感謝いたします。また、日頃より、ご助言をいただきました独立行政法人情報通信研究機構の村上陽平氏に感謝いたします。そして、日頃から多くのご助言とご協力をいただきました石田研究室の皆様にも心より感謝いたします。

## 参考文献

- [1] 野村早恵子, 石田亨, 船越要, 安岡美佳, 山下直美: アジアにおける異文化コラボレーション実験 2002 : 機械翻訳を介したソフトウェア開発, 情報処理, Vol. 44, No. 5 (2003).
- [2] Ishida, T.: Language Grid: An Infrastructure for Intercultural Collaboration, *IEEE/IPSJ Symposium on Applications and the Internet(SAINT-06)*, pp. 96–100 (2006).
- [3] Uchimoto, K., Hayashida, N., Ishida, T., Isahara, H.: Automatic Rating of Machine Translatability, *10th Machine Translation Summit (MT Summit X)*, pp. 235–242 (2005).
- [4] Shigenobu, T.: Evaluation and Usability of Back Translation for Intercultural Communication, *HCI International 2007(HCII-07)*, pp. 259–265 (2007).
- [5] Uchimoto, K., Hayashida, N., Ishida, T., Isahara, H.: Automatic Detection and Semi-Automatic Revision of Non-Machine-Translatable Parts of a Sentence, *International Conference on Language Resources and Evaluation (LREC-06)*, pp. 703–708 (2006).
- [6] 高橋哲朗, 岩倉友哉, 飯田龍, 乾健太郎: KURA: 統一のかつ宣言的記述法に基づく言い換え知識の開発環境, 電子情報通信学会思考と言語研究会 (2001).
- [7] Clarence A. Ellis, Simon J. Gibbs, Gail L. Rein: Groupware: some issues and experiences, *Communications of the ACM*, Vol. 34, No. 1, pp. 38–58 (1991).
- [8] TIDES: Linguistic Data Annotation Specification: Assessment of Fluency and Adequacy in Arabic-English and Chinese-English Translations. (2002). <http://www.lde.upenn.edu/TIDES/Translation/TransAssess02.pdf>.
- [9] 石田亨, 内元清貴, 山下直美, 吉野孝: 機械翻訳を用いた異文化コラボレーション, 情報処理, Vol. 47, No. 3, pp. 269–275 (2006).
- [10] 石田亨, 林田尚子, 野村早恵子: 異文化コラボレーションに向けて 機械翻訳システムの相互作用性, 電子情報通信学会技術研究報告 . AI , 人工知能と知識処理, Vol. 103, No. 244, pp. 37–41 (2003).
- [11] Yamashita, N. and Ishida, T.: Effects of Machine Translation on Collaborative Work, *International Conference on Computer Supported Cooperative Work(CSCW-06)*, pp. 512–523 (2006).

## 付録

### A.1 共同翻訳実験のログ

5.2.1 章の実験において使用した「環境問題」のドメインを持つ文書の2文目「大気汚染は、私たちが日常生活する上で必要なところから行われています。」という原文に対して共同翻訳の各種プロトコルを適用した結果のログを以下に示す。なお、以下に単純な機械翻訳の適用による翻訳文と平均の評価値を示す。  
原文：大気汚染は、私たちが日常生活する上で必要なところから行われています。

翻訳文:When we live daily, air pollution is performed from a necessary place.

平均評価値:3

#### A.1.1 基本プロトコルを適用した場合

原文：大気汚染は、私たちが日常生活する上で必要なところから行われています。

翻訳文:Air pollution mainly occurs in the places we live in.

平均評価値:3.33

表 A.1: 基本プロトコルを適用した場合

原文	翻訳文	折り返し翻訳文
プロトコル開始		
大気汚染は、私たちが日常生活する上で必要なところから行われています。	When we live daily, air pollution is performed from a necessary place.	
翻訳文側修正		
	When we live daily, air pollution is performed from necessary places.	私達が毎日住んでいる時に、大気汚染は必要な場所から実行される。
原文側修正		
私たちが日常生活する上で必要なところを中心に、大気汚染は発生しています。	When we live daily, air pollution occurs centering on a necessary place.	
翻訳文側修正		
	When we live daily, air pollution is performed from necessary places.	私達が毎日住んでいる時に、大気汚染は必要な場所から実行される。
原文側修正		
私たちが日常生活で必要とする場所を中心として、大気汚染は発生しています。	We bring about air pollution centering on a place needed by daily life.	

翻訳文側修正		
	We bring about air pollution centering on places needed by daily life.	私達は日常生活に必要なであった場所の上の大気汚染センタリングを引き起こす。
原文側修正		
私たちが日常生活で必要とする場所において、大気汚染は発生しています。	We bring about air pollution in a place needed by daily life.	
翻訳文側修正		
	We bring about air pollution in places needed by daily life.	日常生活に必要なであった場所で私達は大気汚染を引き起こす。
原文側修正		
大気汚染が発生するのは、主に私たちが日常生活で必要とする場所においてです。	It's in a place we need by daily life mainly that air pollution occurs.	
翻訳文側修正		
	It's in the places we live in that air pollution mainly occurs.	それは、大気汚染が主に起こることにおいて、私達が生きている場所にある。
原文側修正		
大気汚染は、私たちが日常生活で必要とする場所で主に発生します。	We bring about air pollution mainly at a place needed by daily life.	
翻訳文側修正		
	Air pollution mainly occurs in the places we live in.	大気汚染は主に、私達が住んでいるところどころに存在している。
成功でプロトコル終了		

### A.1.2 折り返し翻訳を用いたプロトコルを適用した場合

原文：大気汚染は、私たちが日常生活する上で必要なところから行われています。

翻訳文: Air pollution is generated from something necessary for our daily life.

平均評価値: 4.33

表 A.2: 折り返し翻訳を用いたプロトコルを適用した場合

原文	翻訳文	折り返し翻訳文	翻訳文の折り返し翻訳文
プロトコル開始			
大気汚染は、私たちが日常生活する上で必要なところから行われています。	When we live daily, air pollution is performed from a necessary place.	私達が毎日住んでいる時に、大気汚染は必要な場所から実行される。	

原文側修正			
私たちの日常生活で必要なことから、大気汚染は行われています。	Because it's necessary by our daily life, air pollution is performed.	それが私達の日常生活によって必要なので、大気汚染は実行される。	
原文側修正			
私たちの日常生活で必要なところから、大気が汚染されています。	The atmosphere is polluted from a necessary place by our daily life.	大気は必要な場所から私達の日常生活によって汚染される。	
原文側修正			
私たちの日常生活上で必要なところから、大気汚染が行われています。	Air pollution is performed from a necessary place on our daily life.	大気汚染は私達の日常生活の上の必要な場所から実行される。	
原文側修正			
私たちの日常生活上で必要なものから、大気汚染が行われています。	Air pollution is performed from something necessary on our daily life.	大気汚染は何か私達の日常生活の上で必要なものから実行される。	
翻訳文側修正			
	Air pollution is generated from something necessary on our daily life.	大気汚染は何か私達の日常生活の上で必要なものから発生する。	Air pollution occurs from something necessary on our some daily life.
翻訳文側修正			
	Air pollution is generated from something necessary on our daily life.	大気汚染は何か私達の日常生活の上で必要なものから発生する。	Air pollution occurs from something necessary on our some daily life.
翻訳文側修正			
	Air pollution is generated from something necessary for our daily life.	大気汚染は何か私達の日常生活に必要なものから発生する。	Air pollution occurs from something necessary to our daily life.
成功でプロトコル終了			

### A.1.3 上限，下限の閾値を導入したプロトコルを適用した場合

原文：大気汚染は、私たちが日常生活する上で必要なところから行われています。

翻訳文: Air pollution is being developed from places necessary to our daily living .

平均評価値: 4.33

表 A.3: 上限, 下限の閾値を導入したプロトコルを適用した場合

原文	翻訳文	折り返し翻訳文	翻訳文の折り返し翻訳文
プロトコル開始			
大気汚染は、私たちが日常生活する上で必要なところから行われています。	When we live daily, air pollution is performed from a necessary place.	私達が毎日住んでいる時に、大気汚染は必要な場所から実行される。	
原文側修正			
私たちが日常生活するのに必要なところから、大気汚染は進んでいく。	Air pollution is being developed from a place necessary to our living daily.	大気汚染は毎日私達の生活に必要な場所から発展している。	
翻訳文側修正 (原文側は折り返し翻訳文を原文の意味内容を推定できる程度は良いと評価)			
	Air pollution is being increased in places necessary to our daily living .	大気汚染は私達の毎日の生活に必要な場所で増大している。	Air pollution increases at our location necessary to a daily life.
翻訳文側修正			
	Air pollution is being developed from places necessary to our daily living .	大気汚染は私達の毎日の生活に必要な場所から発展している。	Air pollution is developing from our location necessary to a daily life.
成功でプロトコル終了			

## A.2 ソースコード

### A.2.1 Server.java

```

package collaborative.server;
import java.io.FileInputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.UnknownHostException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.Properties;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Server extends ServerSocket{
protected ArrayList<UserThread> threads = new ArrayList<UserThread>();
protected Properties prop = null;
protected HashMap<String,String> languagePairMap = null;
protected HashMap<Integer,String> workcodeMap = null;
protected TranslationClient translator = null;
protected Logger logger = Logger.getLogger("server.logging");
protected int serviceNumber = 1;
protected final static int PORT = 9876;

```

```

public Server() throws IOException{this(0);}
public Server(int port) throws IOException{
super(port);
this.prop = new Properties();
try{this.prop.loadFromXML(new FileInputStream("Colab_server.properties"));}
catch(IOException e) {e.printStackTrace();}
logger.setLevel(Level.ALL);
}

public void setTranslator(){
try{this.translator = new TranslationClient(this.getProperty("langrid.username"),
this.getProperty("langrid.password"));}
catch(Exception e){e.printStackTrace();}
}
public TranslationClient getTranslator(){
if(this.translator == null)this.setTranslator();
return this.translator;
}
public int getPort(){return this.getLocalPort();}
public InetAddress getHost() throws UnknownHostException{return InetAddress.getLocalHost();}
public void addThread(UserThread thread){this.threads.add(thread);}
public boolean canFindPartner(UserThread t){
for(UserThread userThread : threads)
if(this.isMatched(userThread, t)){
new ServiceController(userThread, t, this, this.serviceNumber);
this.serviceNumber++;
this.threads.remove(userThread);
return true;
}
return false;
}
public boolean isMatched(UserThread t1, UserThread t2){
if(t1.getProtocolNumber() == t2.getProtocolNumber())
if(t1.getUserType().equals("source") && t2.getUserType().equals("target")
&& t1.getPartnerLanguage().equals(t2.getLanguage()))
if(this.isSupportedPair(t1.getLanguage(), t1.getPartnerLanguage())) return true;
else if(t2.getUserType().equals("source") && t1.getUserType().equals("target")
&& t2.getPartnerLanguage().equals(t1.getLanguage()))
if(this.isSupportedPair(t2.getLanguage(), t2.getPartnerLanguage())) return true;
return false;
}
public boolean isSupportedPair(String sourceLang, String targetLang){
if(this.getLanguagePairMap().get(sourceLang).contains(targetLang)) return true;
else return false;
}
public String getProperty(String property){return this.prop.getProperty(property);}
public HashMap<String,String> getLanguagePairMap(){
if(this.languagePairMap == null){
this.languagePairMap = new HashMap<String,String>();
String supported = this.getProperty("language.pair");
ArrayList<String> supportedPair = new ArrayList<String>(Arrays.asList(supported.split(";")));
for(String s : supportedPair){String[] keyval = s.split(":");
this.languagePairMap.put(keyval[0], keyval[1]);}
}
}

```

```

}
return this.languagePairMap;
}
public void setWorkcodeMap(){
String[] workcodes = this.getProperty("workcode.codes").split(",");
this.workcodeMap = new HashMap<Integer,String>();
for(String codeStr : workcodes){
int code = Integer.parseInt(codeStr);
this.workcodeMap.put(code, this.getProperty("workcode."+code));
}
}
public HashMap<Integer,String> getWorkcodeMap(){
if(this.workcodeMap == null)this.setWorkcodeMap();
return this.workcodeMap;
}
public Logger getLogger(){return this.logger;}
public static void main(String[] args) {
Server colabServer = null;
try{
colabServer = new Server(Server.PORT);
colabServer.getLogger().log(Level.INFO, "Start Server! HOST: "+colabServer.getHost()+
" , PORT: "+colabServer.getPort());
while(true){
colabServer.getLogger().log(Level.INFO, colabServer.threads.size()+" threads are waiting.");
UserThread t = new UserThread(colabServer.accept(), colabServer);
if(!colabServer.canFindPartner(t)) colabServer.addThread(t);
}
} catch (UnknownHostException e){e.printStackTrace();
} catch (IOException e){e.printStackTrace();
} finally {
try{colabServer.getLogger().log(Level.INFO, "Socket of this Server is closing");colabServer.close();}
catch(IOException e){e.printStackTrace();}
}
}
}

```

## A.2.2 ServiceController.java

```

package collaborative.server;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.lang.reflect.Array;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Hashtable;
import java.util.logging.Level;
import java.util.regex.*;
import collaborative.common.Pair;
import org.apache.poi.hssf.usermodel.HSSFCell;

```



```

import org.apache.poi.hssf.usermodel.HSSFCellStyle;
import org.apache.poi.hssf.usermodel.HSSFRichTextString;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.hssf.util.Region;

public class ServiceController{
protected Server coreServer = null;
protected Pair<UserThread, UserThread> userPair = null;
protected ArrayList<SentenceTask> tasks = new ArrayList<SentenceTask>();
protected int maxNumberOfCycles = 0;
protected int protocolNumber = 0;
protected int serviceId = -1;
protected String startTime = null;
protected Hashtable<String,Boolean> shareMap = new Hashtable<String,Boolean>();
protected final Pair[] ESCAPE = {new Pair<String, String>(".", "[[#dot]]"),
new Pair<String, String>("!", "[[#exclamation]]"),new Pair<String, String>("?", "[[#question]]")};
protected final String[] SEPARATOR = {".", "?", "!", " ", "。", " ", "!", " ?"};
protected final String LINE_FEED = "<<linefeed>>";

public ServiceController(UserThread t1, UserThread t2, Server coreServer, int serviceNumber){
super();
if(t1.getUserType().equals("source")) this.userPair = new Pair<UserThread, UserThread>(t1, t2);
else this.userPair = new Pair<UserThread, UserThread>(t2, t1);
this.serviceId = serviceNumber;
this.coreServer = coreServer;
this.maxNumberOfCycles = Integer.parseInt(this.coreServer.getProperty("max_cycles"));
this.userPair.getFirst().setController(this);
this.userPair.getSecond().setController(this);
this.userPair.getFirst().start();
this.userPair.getSecond().start();
startTime = this.getStringTime();
}

public Server getCollaborationServer(){return this.coreServer;}
public UserThread getSourceUser(){return this.userPair.getFirst();}
public UserThread getTargetUser(){return this.userPair.getSecond();}
public void setOriginalFlag(boolean isSetOriginal){this.shareMap.put("original", isSetOriginal);}
public boolean isSetOriginal(){
if(!this.shareMap.containsKey("original")) this.shareMap.put("original", false);
return this.shareMap.get("original");
}
public String getOriginal(int id){
String ret = null;
do{
try{Thread.sleep(100);} catch(InterruptedException e){e.printStackTrace();}
if(this.isSetOriginal()) ret = this.tasks.get(id).getOriginal(0);
}while(ret == null);
return ret;
}
public String getModifiedOriginal(int id){
SentenceTask task = this.tasks.get(id);String temp = null;int timeout = 0;

```

```

do{
for(int i=task.getNumberOfTuples()-1;i>=0;i--)
if((temp = task.getModifiedOriginal(i)) != null) return temp;
try{Thread.sleep(100);} catch(InterruptedException e){e.printStackTrace();} timeout++;
}while(timeout < 300);
return temp;
}

public String getTranslation(int id){
SentenceTask task = this.tasks.get(id);String temp = null;int timeout = 0;
do{
for(int i=task.getNumberOfTuples()-1;i>=0;i--)
if(task.getModifiedOriginal(i) != null){
do{
try{Thread.sleep(100);} catch(InterruptedException e){e.printStackTrace();}
if((temp = task.getTranslation(i)) != null) break;timeout++;
}while(timeout < 300);
break;
}
if(temp != null) break;
try{Thread.sleep(100);} catch(InterruptedException e){e.printStackTrace();} timeout++;
}while(timeout < 300);
return temp;
}

public String getModifiedTranslation(int id){
String ret = null;SentenceTask task = this.tasks.get(id);int timeout = 0;
do{
for(int i=task.getNumberOfTuples()-1;i>=0;i--)
if(!task.isEmpty(i)){
do{
try{Thread.sleep(100);} catch(InterruptedException e){e.printStackTrace();}
ret = task.getModifiedTranslation(i);timeout++;
}while(ret == null && timeout < 300);
return ret;
}
try{Thread.sleep(100);} catch(InterruptedException e){e.printStackTrace();}
timeout++;
}while(timeout < 300);
return ret;
}

public String getBackTranslation(int id){
String ret = null;SentenceTask task = this.tasks.get(id);int timeout = 0;
do{
for(int i=task.getNumberOfTuples()-1;i>=0;i--)
if(!task.isEmpty(i)){
do{
try{Thread.sleep(100);} catch(InterruptedException e){e.printStackTrace();}
ret = task.getBackTranslation(i);timeout++;
}while(ret == null && timeout < 300);
return ret;
}
try{Thread.sleep(100);} catch(InterruptedException e){e.printStackTrace();}timeout++;
}while(timeout < 300);
return ret;
}

```

```

}
public String getReBackTranslation(int id){
String ret = null;SentenceTask task = this.tasks.get(id);int timeout = 0;
do{
for(int i=task.getNumberOfTuples()-1;i>=0;i--){
if(!task.isEmpty(i)){
do{
try{Thread.sleep(100);} catch(InterruptedException e){e.printStackTrace();}
ret = task.getReBackTranslation(i);timeout++;
}while(ret == null && timeout < 300);
return ret;
}
try{Thread.sleep(100);} catch(InterruptedException e){e.printStackTrace();}timeout++;
}while(timeout < 300);
return ret;
}
}
public int getProtocolNumber(){
if(this.protocolNumber == 0)this.protocolNumber = this.getSourceUser().getProtocolNumber();
return this.protocolNumber;
}
public void setModifiedOriginal(int id, String str){this.tasks.get(id).setModifiedOriginal(str);}
public void setModifiedTranslation(int id, String str){this.tasks.get(id).setModifiedTranslation(str);}
public void setPartnerWorkCode(int id, int code, String myUserType){
this.tasks.get(id).setWorkCode(code);
if(myUserType.equals("source")) this.getTargetUser().setWorkCode(id, code);
else if(myUserType.equals("target")) this.getSourceUser().setWorkCode(id, code);
if(this.isFinished())
this.coreServer.getLogger().log(Level.INFO, "No."+this.serviceId+" service is Finished.");
}
public boolean isFinished(){
for(SentenceTask t : this.tasks) if(t.getWorkCode() - 100*this.getProtocolNumber() < 20) return false;
return true;
}
}
public void setAllSentences(String sentences){
this.setTasks(sentences);
for(int j=0;j<this.tasks.size();j++)
if(this.getProtocolNumber() == 1) this.tasks.get(j).setWorkCode(111);
else if(this.getProtocolNumber() == 2) this.tasks.get(j).setWorkCode(201);
else if(this.getProtocolNumber() == 3) this.tasks.get(j).setWorkCode(303);
this.setInitWorkCodesToClients();
}
}
public String encodeURL(String text){
return this.encodeFromRegex(text, "\\s?https?:://[-_!.~*'()a-zA-Z0-9;\\|/?:@&=+,%#]+");
}
}
public String encodeDecimalNumber(String text){return this.encodeFromRegex(text, "\\d*\\.\\d+");}
public ArrayList<String> getArrayFromRegex(String text, String regex){
ArrayList<String> urlArray = new ArrayList<String>();
Matcher matcher = Pattern.compile(regex).matcher(text);
while(matcher.find())urlArray.add(text.substring(matcher.start(), matcher.end()));
return urlArray;
}
}
public String encodeFromRegex(String text, String regex){
ArrayList<String> regexArray = this.getArrayFromRegex(text,regex);

```

```

for(int i=0;i<regexArray.size();i++){
int index = text.indexOf(regexArray.get(i));
String textHead = text.substring(0,index);
String textTail = text.substring(index+regexArray.get(i).length());
String substr = regexArray.get(i);
for(Pair p : this.ESCAPE)substr = substr.replaceAll((String)p.getFirst(), (String)p.getSecond());
text = textHead + substr + textTail;
}
return text;
}
public String encodeExceptionCharacter(String text){
ArrayList<String> exceptionWords = null;
for(Pair symbolPair : this.ESCAPE){
exceptionWords = this.getExceptionWords((String)symbolPair.getFirst());int head=0, tail = 0;
while((tail = text.indexOf(' ',head)) != -1){
if(exceptionWords.contains(text.substring(head,tail)))
if(head == 0 || text.substring(head-1).startsWith(" ")){
String substr = text.substring(head,tail);
String textHead = text.substring(0,head);
String textTail = text.substring(tail);
substr = substr.replaceAll((String)symbolPair.getFirst(), (String)symbolPair.getSecond());
text = textHead+substr+textTail;
}
head = tail+1;
}
if(exceptionWords.contains(text.substring(head)))
if(head == 0 || text.substring(head-1).startsWith(" ")){
String substr = text.substring(head);
String textHead = text.substring(0,head);
substr = substr.replaceAll((String)symbolPair.getFirst(), (String)symbolPair.getSecond());
if(head==0) text = substr;
else text = textHead+substr;
}
}
return text;
}
public String decodeExceptionCharacter(String text){
for(Pair p : this.ESCAPE) text = text.replaceAll((String)p.getSecond(), (String)p.getFirst());
return text;
}
public ArrayList<Pair<String, Boolean>> getSentenceArray(String text){
text = this.encodeExceptionCharacter(this.encodeDecimalNumber(this.encodeURL(text)));
ArrayList<ArrayList<String>> sentenceArray = new ArrayList<ArrayList<String>>();
sentenceArray.add(new ArrayList<String>());
sentenceArray.get(0).add(text);
for(String separator : this.SEPARATOR){
for(int j=0;j<sentenceArray.size();j++){
sentenceArray.set(j, new ArrayList<String>(Arrays.asList(sentenceArray.get(j).get(0).split("\\\\"+separator))));
if(sentenceArray.get(j).size() > 1)
if(sentenceArray.get(j).get(sentenceArray.get(j).size()-1).equals("")) {
sentenceArray.get(j).remove(sentenceArray.get(j).size()-1);
for(int k=0;k<sentenceArray.get(j).size();k++)
sentenceArray.get(j).set(k, sentenceArray.get(j).get(k)+separator);
}
}
}
}

```

```

} else
for(int k=0;k<sentenceArray.get(j).size()-1 ;k++)
sentenceArray.get(j).set(k, sentenceArray.get(j).get(k) + separator);
}
sentenceArray = this.flattenDim(sentenceArray);
}
ArrayList<String> newSentenceArray = this.flatten(sentenceArray);
ArrayList<Boolean> lineFeedArray = new ArrayList<Boolean>();
for(int i=0;i<newSentenceArray.size();i++)
if(newSentenceArray.get(i).startsWith(this.LINE_FEED)){
lineFeedArray.add(true);
newSentenceArray.set(i, newSentenceArray.get(i).substring(this.LINE_FEED.length()));
} else lineFeedArray.add(false);
if(newSentenceArray.get(newSentenceArray.size()-1).equals("")){
lineFeedArray.remove(newSentenceArray.size()-1);sentenceArray.remove(newSentenceArray.size()-1);
}
for(int i=0;i<newSentenceArray.size();i++)
newSentenceArray.set(i, newSentenceArray.get(i).replaceAll("\\Q"+this.LINE_FEED+"\\E", ""));
while(true){
boolean flag = false;int i = 0;
for(;i<newSentenceArray.size();i++) if(newSentenceArray.get(i).matches("\\s*")){flag = true;break;}
if(flag) newSentenceArray.remove(i); else break;
}
for(int j=0;j<newSentenceArray.size();j++)
newSentenceArray.set(j, this.decodeExceptionCharacter(newSentenceArray.get(j)));
ArrayList<Pair<String,Boolean>> retArray = new ArrayList<Pair<String,Boolean>>();
for(int i=0;i<newSentenceArray.size();i++)
retArray.add(new Pair<String, Boolean>(newSentenceArray.get(i), lineFeedArray.get(i)));
return retArray;
}

public ArrayList<String> getExceptionWords(String symbol){
String[] retArray = {}, alphabet = {"a","b","c","d","e","f","g","h","i","j","k","l","m",
"n","o","p","q","r","s","t","u","v","w","x","y","z"};
for(int i=0;i<alphabet.length;i++){
String temp = this.coreServer.getProperty("exception"+symbol+alphabet[i]);
if(temp == null) continue;
retArray = (String[])this.concat(String.class, retArray, temp.split(","));
}
return new ArrayList<String>(Arrays.asList(retArray));
}

public void notifyPartnerExit(String userType){
this.createLogFile();
if(userType.equals("source"))this.getTargetUser().setReset();
else if(userType.equals("target")) this.getSourceUser().setReset();
}

public Object[] concat(Class cls, Object[] a, Object[] b) {
Object[] dst = (Object[])Array.newInstance(cls, a.length + b.length);
System.arraycopy(a, 0, dst, 0, a.length);System.arraycopy(b, 0, dst, a.length, b.length);
return dst;
}

public ArrayList<ArrayList<String>> flattenDim(ArrayList<ArrayList<String>> list){
for(int i=list.size()-1;i>=0;i--){

```

```

ArrayList<String> listElem = list.get(i);
for(int j=0;j<listElem.size();j++)list.add(i+j, new ArrayList<String>(Arrays.asList(listElem.get(j))));
list.remove(listElem);
}
return list;
}
public ArrayList<String> flatten(ArrayList<ArrayList<String>> list){
ArrayList<String> newList = new ArrayList<String>();
for(int i=0;i<list.size();i++) for(int j=0;j<list.get(i).size();j++)newList.add(list.get(i).get(j));
return newList;
}
public void translateLastTuple(int id){
new TranslationThread(this.tasks.get(id).lastTuple(), this.tasks.get(id), this);}
public void setTasks(String original){
ArrayList<Pair<String,Boolean>> stringPairList = this.getSentenceArray(original);
boolean isLast = false;
for(int i=0;i<stringPairList.size();i++){
SentenceTask tempTask =
new SentenceTask(stringPairList.get(i).getFirst(), stringPairList.get(i).getSecond(), i);
this.tasks.add(tempTask);
if(i == stringPairList.size()-1) isLast = true;
new TranslationThread(this.tasks.get(i).getTuple(0), tempTask, this, true, isLast);
}
}
public void setInitWorkCodesToClients(){
String mes = "";
for(SentenceTask t : this.tasks) mes += t.getWorkCode()+" ";
mes = mes.substring(0, mes.length()-1);
this.getSourceUser().setInitWorkCodes(mes);this.getTargetUser().setInitWorkCodes(mes);
}

public String getInitTasks(String userType){
String ret = "";
while(!this.isSetOriginal()) try{Thread.sleep(100);}catch(InterruptedException e){e.printStackTrace();}
if(userType.equals("source")) for(int i=0;i<this.tasks.size();i++)
ret += "<<" + i + "#" + this.tasks.get(i).getWorkCode() + "#" + this.getOriginal(i) + ">>";
else if(userType.equals("target")) for(int i=0;i<this.tasks.size();i++)
ret += "<<" + i + "#" + this.tasks.get(i).getWorkCode() + "#" + this.getTranslation(i) + ">>";
return ret;
}

public void createLogFile(){
String filePath = "log_"+this.getStringDate()+"_"+this.serviceId+".xls";
File directory = new File("log");
if(!directory.exists()) directory.mkdir();File workFile = new File(directory,filePath);
if(!workFile.exists()) try {workFile.createNewFile();} catch (IOException e) {e.printStackTrace();}
HSSFWorkbook workBook = new HSSFWorkbook();
HSSFCellStyle style = workBook.createCellStyle();
style.setWrapText(true);
HSSFSheet resultSheet = workBook.createSheet("Result Sentence");
resultSheet.setColumnWidth((short)0, (short)18000);resultSheet.setColumnWidth((short)1, (short)18000);
String[][] property = {
{"Protocol", Integer.toString(this.getProtocolNumber())},

```

```

{"Language", this.getSourceUser().getLanguage()+" -> "+this.getTargetUser().getLanguage()},
{"SL-part User("+this.getSourceUser().getLanguage()+")",
this.getSourceUser().sock.getRemoteSocketAddress().toString(),
"TL-part User("+this.getTargetUser().getLanguage()+")",
this.getTargetUser().sock.getRemoteSocketAddress().toString()},
{"Time", this.startTime+" - "+this.getStringTime()}
};
for(int j=0;j<property.length;j++){
HSSFRow row = resultSheet.createRow(j);
for(int k=0;k<property[j].length;k++){
HSSFCell cell = row.createCell((short)k);
cell.setCellValue(new HSSFRichTextString(property[j][k]));
}
}
for(int j=0;j<this.tasks.size();j++){
HSSFRow row = resultSheet.createRow(property.length+1+j);
HSSFCell cellOrig = row.createCell((short)0);
cellOrig.setCellValue(new HSSFRichTextString(this.getOriginal(j)));
cellOrig.setCellStyle(style);
HSSFCell cellTrans = row.createCell((short)1);
cellTrans.setCellValue(new HSSFRichTextString(this.getModifiedTranslation(j)));
cellTrans.setCellStyle(style);
if(this.tasks.get(j).getWorkCode()-100*this.getProtocolNumber() == 21){
HSSFCell cellBatsu = row.createCell((short)2);
cellBatsu.setCellValue(new HSSFRichTextString("×"));
cellBatsu.setCellStyle(style);
}
}
for(int i=0;i<this.tasks.size();i++){
HSSFSheet sheet = workbook.createSheet("Sentence - "+(i+1));
sheet.setColumnWidth((short)0, (short)2500);
sheet.setColumnWidth((short)1, (short)2700);
for(int j=2;j<=7;j++) sheet.setColumnWidth((short)j, (short)6100);
style.setBorderBottom(HSSFCellStyle.BORDER_THICK);
HSSFRow row = sheet.createRow(0);
String[] titles = {"Date", "Work Code", "Original", "Modified Original", "Translation",
"Modified Translation", "Back Translation", "Back Translation Of Translation"};
for(int j=0;j<titles.length;j++){
HSSFCell cell = row.createCell((short)j);
cell.setCellValue(new HSSFRichTextString(titles[j]));
cell.setCellStyle(style);
}
SentenceTask task = this.tasks.get(i);
style.setBorderBottom(HSSFCellStyle.BORDER_NONE);
HashMap<Integer,String> workcodeMap = this.coreServer.getWorkcodeMap();
for(int j=0;j<task.getNumberOfTuples();j++){
row = sheet.createRow(j+1);
HSSFCell calCell = row.createCell((short)0);
calCell.setCellValue(new HSSFRichTextString(task.getStringTime(j)));
HSSFCell workCell = row.createCell((short)1);
if(task.getWorkCode(j) != 0){
workCell.setCellValue(new HSSFRichTextString(workcodeMap.get(task.getWorkCode(j))));
workCell.setCellStyle(style);
}
}

```

```

sheet.addMergedRegion(new Region(j+1, (short)1, j+1, (short)7));
}
HSSFCell oriCell = row.createCell((short)2);
if(task.getOriginal(j) != null){
oriCell.setCellValue(new HSSFRichTextString(task.getOriginal(j)));
oriCell.setCellStyle(style);
}
HSSFCell modoriCell = row.createCell((short)3);
if(task.getModifiedOriginal(j) != null){
modoriCell.setCellValue(new HSSFRichTextString(task.getModifiedOriginal(j)));
modoriCell.setCellStyle(style);
}
HSSFCell traCell = row.createCell((short)4);
if(task.getTranslation(j) != null){
traCell.setCellValue(new HSSFRichTextString(task.getTranslation(j)));
traCell.setCellStyle(style);
}
HSSFCell modtraCell = row.createCell((short)5);
if(task.getModifiedTranslation(j) != null){
modtraCell.setCellValue(new HSSFRichTextString(task.getModifiedTranslation(j)));
modtraCell.setCellStyle(style);
}
HSSFCell backCell = row.createCell((short)6);
if(task.getBackTranslation(j) != null){
backCell.setCellValue(new HSSFRichTextString(task.getBackTranslation(j)));
backCell.setCellStyle(style);
}
HSSFCell rebackCell = row.createCell((short)7);
if(task.getReBackTranslation(j) != null){
rebackCell.setCellValue(new HSSFRichTextString(task.getReBackTranslation(j)));
rebackCell.setCellStyle(style);
}
}
}
}
FileOutputStream fileOut =null;
try {fileOut = new FileOutputStream("log\\"+filePath);workBook.write(fileOut);fileOut.close();
} catch (FileNotFoundException e) {e.printStackTrace();
} catch (IOException e) {e.printStackTrace();}
}
public String getStringDate(){
SimpleDateFormat sdf = new SimpleDateFormat("yyyyMMddHHmmss");
return sdf.format(Calendar.getInstance().getTime());
}
public String getStringTime(){
SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
return sdf.format(Calendar.getInstance().getTime());
}
}
}

```

### A.2.3 UserThread.java

```

package collaborative.server;
import java.io.BufferedReader;
import java.io.IOException;

```



```

import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
import java.net.SocketException;
import java.util.logging.Level;

public class UserThread extends Thread {
protected Socket sock = null;
protected Server serverCore = null;
protected ServiceController controller = null;
protected BufferedReader in = null;
protected PrintWriter out = null;
protected int protocolNumber = 0;
protected String userType = null;
protected String language = null;
protected String partnerLanguage = null;

public UserThread(Socket sock, Server serverCore){
super();this.sock = sock;this.serverCore = serverCore;
try{
this.in = new BufferedReader(new InputStreamReader(this.sock.getInputStream()));
this.out = new PrintWriter(this.sock.getOutputStream(), true);
this.initiate(this.in.readLine());
} catch(IOException e) {e.printStackTrace();}
}

public void initiate(String firstMessage){
String[] mesArray = firstMessage.split(" ");
this.userType = mesArray[0];
this.protocolNumber = Integer.parseInt(mesArray[1]);
this.language = mesArray[2];
if(this.userType.equals("source")){
this.partnerLanguage = mesArray[3];
this.serverCore.getLogger().log(Level.INFO, "Source user logged in. [Protocol : "+
this.getProtocolNumber()+"] [Language : "+this.getLanguage()+" -> "+
this.getPartnerLanguage()+"] [Host : "+this.sock.getRemoteSocketAddress()+"]");
} else
this.serverCore.getLogger().log(Level.INFO, "Target user logged in. [Protocol : "+this.getProtocolNumber()
+"] [Language : "+this.getLanguage()+"] [Host : "+this.sock.getRemoteSocketAddress()+"]");
}

public int getProtocolNumber(){return this.protocolNumber;}
public String getUserType(){return this.userType;}
public String getLanguage(){return this.language;}
public String getPartnerLanguage(){return this.partnerLanguage;}
public ServiceController getController(){return this.controller;}
public String getHost(){return this.sock.getLocalAddress().toString();}
public int getPort(){return this.sock.getLocalPort();}
public void setProtocolNumber(int protocolNumber){this.protocolNumber = protocolNumber;}
public void setUserType(String userType){this.userType = userType;}
public void setLanguage(String language){this.language = language;}
public void setPartnerLanguage(String partnerLanguage){this.partnerLanguage = partnerLanguage;}
public void setController(ServiceController cont){this.controller = cont;}

```

```

public void setWorkCode(int id, int workCode){this.send(203, id, String.valueOf(workCode));}
public void setTasks(String tasks){this.send(204, 0, tasks);}
public void setInitWorkCodes(String codes){this.send(205, 0, codes);}
public void setOriginal(int id, String sentence){this.send(210, id, sentence);}
public void setModifiedOriginal(int id, String sentence){this.send(211, id, sentence);}
public void setTranslation(int id, String sentence){this.send(212, id, sentence);}
public void setModifiedTranslation(int id, String sentence){this.send(213, id, sentence);}
public void setBackTranslation(int id, String sentence){this.send(214, id, sentence);}
public void setReBackTranslation(int id, String sentence){this.send(215, id, sentence);}
public void setModifiedTranslationForConfirmation(int id, String sentence){this.send(216, id, sentence);}
public void setBackTranslationInModifying(int id, String sentence){this.send(217, id, sentence);}
public void setReBackTranslationInModifying(int id, String sentence){this.send(218, id, sentence);}
public void setModifiedTranslationFirst(int id, String sentence){this.send(219, id, sentence);}
public void setModifiedTranslationNotFirst(int id, String sentence){this.send(220, id, sentence);}
public void setReset(){this.send(900, 0, "");}
public boolean decode(String clientMessage){
    if(clientMessage.matches("[0-9]{7}.*")){
        int code = Integer.parseInt(clientMessage.substring(0, 3));
        int id = Integer.parseInt(clientMessage.substring(3, 7));
        switch(code){
            case 104:this.setTasks(this.getController().getInitTasks(this.userType));break;
            case 110:this.setOriginal(id, this.getController().getOriginal(id));break;
            case 111:this.setModifiedOriginal(id, this.getController().getModifiedOriginal(id));break;
            case 112:this.setTranslation(id, this.getController().getTranslation(id));break;
            case 113:this.setModifiedTranslation(id, this.getController().getModifiedTranslation(id));break;
            case 114:this.setBackTranslation(id, this.getController().getBackTranslation(id));break;
            case 115:this.setReBackTranslation(id, this.getController().getReBackTranslation(id));break;
            case 116:
                this.setModifiedTranslationForConfirmation(id, this.getController().getModifiedTranslation(id));break;
            case 117:this.setBackTranslationInModifying(id, this.getController().getBackTranslation(id));break;
            case 118:this.setReBackTranslationInModifying(id, this.getController().getReBackTranslation(id));break;
            case 119:this.setModifiedTranslationFirst(id, this.getController().getModifiedTranslation(id));break;
            case 120:this.setModifiedTranslationNotFirst(id, this.getController().getModifiedTranslation(id));break;
            case 200:this.getController().setAllSentences(clientMessage.substring(7));break;
            case 203:this.getController().setPartnerWorkCode(id, Integer.parseInt(clientMessage.substring(7)),
                this.getUserType());break;
            case 211:this.getController().setModifiedOriginal(id, clientMessage.substring(7));
                this.getController().translateLastTuple(id);break;
            case 213:this.getController().setModifiedTranslation(id, clientMessage.substring(7));
                this.getController().translateLastTuple(id);break;
            case 900:this.getController().notifyPartnerExit(this.getUserType());return true;
        }
    }
    return false;
}
public void notifyConnection(){
    this.send(901, 0, Integer.toString(this.getController().maxNumberOfCycles));}

public void send(int code, int sentenceNumber, String message){
    String numberStr = String.valueOf(sentenceNumber);
    for(int i=numberStr.length();i<4;i++)numberStr = "0"+numberStr;
    this.out.println(code+numberStr+message);
    this.controller.coreServer.logger.log(Level.INFO,

```

```

"Send message to "+this.getUserType()+" client: "+message);
}

public void run(){
this.controller.coreServer.logger.log(Level.INFO,
this.getUserType()+" User: Thread start.");
this.notifyConnection();
while(true){
try{
String str = this.in.readLine();
this.controller.coreServer.logger.log(Level.INFO, "Receive message from "+this.getUserType()+" client: "+str);
this.decode(str);
} catch(SocketException e){
try{this.sock.close();this.in.close();this.out.close();} catch(IOException e1){e1.printStackTrace();}
this.getController().notifyPartnerExit(this.getUserType());
this.controller.coreServer.logger.log(Level.INFO, this.getUserType()+" User: Thread shutdown.");
break;
} catch(IOException e){e.printStackTrace();}
}
}
}
}

```

#### **A.2.4 TranslationThread.java**

```

package collaborative.server;
import javax.xml.rpc.ServiceException;
import jp.go.nict.langrid.language.Language;
import jp.go.nict.langrid.language.InvalidLanguageTagException;
import jp.go.nict.langrid.ws_1_2.UnknownException;

public class TranslationThread extends Thread {
protected ServiceController cont = null;
protected SentenceTask task = null;
protected InformationTuple tuple = null;
protected Language sourceLang = null;
protected Language targetLang = null;
private boolean isInit = false;
private boolean isLast = false;

public TranslationThread(InformationTuple tuple, SentenceTask task,
ServiceController cont, boolean isInit, boolean isLast){
super();
this.tuple = tuple;this.task = task;this.cont = cont;this.isInit = isInit;this.isLast = isLast;
try{
this.sourceLang = Language.parse(this.cont.getSourceUser().getLanguage());
this.targetLang = Language.parse(this.cont.getTargetUser().getLanguage());
} catch(InvalidLanguageTagException e){e.printStackTrace();}
this.start();
}

public TranslationThread(InformationTuple tuple, SentenceTask task, ServiceController cont){
super();this.tuple = tuple;this.task = task;this.cont = cont;
try{
this.sourceLang = Language.parse(this.cont.getSourceUser().getLanguage());

```

```

this.targetLang = Language.parse(this.cont.getTargetUser().getLanguage());
} catch(InvalidLanguageTagException e){e.printStackTrace();}
this.start();
}

public void sendTaskToSource(){
this.cont.getSourceUser().setWorkCode(this.task.getId(), this.tuple.getWorkCode());}
public void sendTaskToTarget(){
this.cont.getTargetUser().setWorkCode(this.task.getId(), this.tuple.getWorkCode());}
public void run(){
TranslationClient translator = null;
try{
translator = this.cont.coreServer.getTranslator();
if(this.tuple.hasOriginal()) this.tuple.setModifiedOriginal(this.tuple.getOriginal());
if(this.tuple.hasModifiedOriginal()) this.tuple.setTranslation(translator.translate(
sourceLang, targetLang, this.tuple.getModifiedOriginal()));
if(this.tuple.hasTranslation()) this.tuple.setModifiedTranslation(this.tuple.getTranslation());
this.tuple.setBackTranslation(translator.translate(targetLang, sourceLang,
this.tuple.getModifiedTranslation()));
if(!this.tuple.hasModifiedOriginal() && (this.cont.getProtocolNumber() != 1))
this.tuple.setReBackTranslation(translator.translate(sourceLang, targetLang,
this.tuple.getBackTranslation()));
} catch(UnknownException e){
try {
translator.setAlternativeTranslator();new TranslationThread(tuple, task, cont, isInit, isLast);
} catch (ServiceException e1) {e1.printStackTrace();}
} catch(Exception e){e.printStackTrace();}
if(this.isLast){this.cont.setOriginalFlag(true);}
}
}

```

## A.2.5 SentenceTask.java

```

package collaborative.server;
import java.util.ArrayList;

public class SentenceTask {
protected ArrayList<InformationTuple> tuples = new ArrayList<InformationTuple>();
protected boolean hasLineFeed = false;
protected int taskId = -1;
protected int workCode = -1;

public SentenceTask(String original, boolean hasLineFeed, int taskId){
this.hasLineFeed = hasLineFeed;this.taskId = taskId;this.createTuple(original);
}

public String getOriginal(int number){return this.tuples.get(number).getOriginal();}
public String getModifiedOriginal(int number){return this.tuples.get(number).getModifiedOriginal();}
public String getTranslation(int number){return this.tuples.get(number).getTranslation();}
public String getModifiedTranslation(int number){return this.tuples.get(number).getModifiedTranslation();}
public String getBackTranslation(int number){return this.tuples.get(number).getBackTranslation();}
public String getReBackTranslation(int number){return this.tuples.get(number).getReBackTranslation();}
public int getWorkCode(int number){return this.tuples.get(number).getWorkCode();}
public int getWorkCode(){return this.workCode;}
}

```

```

public int getId(){return this.taskId;}
public void setWorkCode(int code){
this.workCode = code;this.createTuple();this.lastTuple().setWorkCode(code);
}
public void setModifiedOriginal(String modifiedOriginal){
this.createTuple();this.lastTuple().setModifiedOriginal(modifiedOriginal);
}
public void setModifiedTranslation(String modifiedTranslation){
this.createTuple();this.lastTuple().setModifiedTranslation(modifiedTranslation);
}
public InformationTuple getTuple(int number){return this.tuples.get(number);}
public void createTuple(){tuples.add(new InformationTuple());}
public void createTuple(String original){tuples.add(new InformationTuple(original));}
public void addTuple(InformationTuple tuple){tuples.add(tuple);}
public InformationTuple lastTuple(){return tuples.get(tuples.size()-1);}
public int getNumberOfTuples(){return this.tuples.size();}
public boolean isEmpty(int number){return this.tuples.get(number).isEmpty();}
public String getStringTime(int number){return this.tuples.get(number).getDate();}
}

```

## A.2.6 InformationTuple.java

```

package collaborative.server;
import java.text.SimpleDateFormat;
import java.util.Calendar;

public class InformationTuple {
protected String original = null;
protected String modifiedOriginal = null;
protected String translation = null;
protected String modifiedTranslation = null;
protected String backTranslation = null;
protected String reBackTranslation = null;
protected int workCode = 0;
protected String date = null;

public InformationTuple(){this.date = this.getStringTime();}
public InformationTuple(String original){this();this.original = original;}

public String getOriginal(){return this.original;}
public String getModifiedOriginal(){return this.modifiedOriginal;}
public String getTranslation(){return this.translation;}
public String getModifiedTranslation(){return this.modifiedTranslation;}
public String getBackTranslation(){return this.backTranslation;}
public String getReBackTranslation(){return this.reBackTranslation;}
public String getDate(){return this.date;}
public int getWorkCode(){return this.workCode;}
public void setOriginal(String original){this.original=original;}
public void setModifiedOriginal(String modifiedOriginal){this.modifiedOriginal=modifiedOriginal;}
public void setTranslation(String translation){this.translation=translation;}
public void setModifiedTranslation(String modifiedTranslation){
this.modifiedTranslation=modifiedTranslation;}
public void setBackTranslation(String backTranslation){this.backTranslation=backTranslation;}
public void setReBackTranslation(String reBackTranslation){this.reBackTranslation=reBackTranslation;}
}

```

```

public void setWorkCode(int workCode){this.workCode=workCode;}
public boolean hasOriginal(){return this.original != null;}
public boolean hasModifiedOriginal(){return this.modifiedOriginal != null;}
public boolean hasTranslation(){return this.translation != null;}
public boolean isEmpty(){
if((this.original == null) && (this.modifiedOriginal == null) && (this.translation == null) &&
(this.modifiedTranslation == null) && (this.reBackTranslation == null) && (this.backTranslation == null))
return true;
return false;
}
public String getStringTime(){
SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
return sdf.format(Calendar.getInstance().getTime());
}
}
}

```

### A.2.7 TranslationClient.java

```

package collaborative.server;
import javax.xml.rpc.ServiceException;
import jp.ac.kyoto_u.i.soc.ai.langrid.langrid_service_1_2.services.JServer.KyotoUJServerLocator;
import jp.ac.kyoto_u.i.soc.ai.langrid.langrid_service_1_2.services.JServer.Translation;
import jp.go.nict.langrid.langrid_service_1_2.services.JServer.NICTJServerLocator;
import jp.go.nict.langrid.language.Language;
import org.apache.axis.client.Stub;

public class TranslationClient {
private Translation server = null;

public TranslationClient() throws ServiceException{
System.setProperty("http.proxyHost", "proxy.kuins.net");System.setProperty("http.proxyPort", "8080");
KyotoUJServerLocator locator = new KyotoUJServerLocator(); this.server = locator.getTranslation();
}

public TranslationClient(String username, String password) throws ServiceException{
this();this.setPassword(password);this.setUsername(username);}

public void setAlternativeTranslator() throws ServiceException{
NICTJServerLocator locator = new NICTJServerLocator(); // 1
this.server = (jp.ac.kyoto_u.i.soc.ai.langrid.langrid_service_1_2.services.JServer.Translation)
locator.getTranslation();
}

public void setUsername(String username){((Stub)this.server).setUsername(username);}
public void setPassword(String password){((Stub)this.server).setPassword(password);}
public String translate(Language source, Language target, String text) throws Exception {
return server.translate(source.getCode(), target.getCode(), text);}
}
}

```

### A.2.8 Client.java

```

package collaborative.client;
import java.awt.Container;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;

```

```

import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ConnectException;
import java.net.Socket;
import java.util.Arrays;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Properties;
import java.util.logging.Level;
import java.util.logging.Logger;
import org.apache.poi.hwpf.HWPFDocument;
import org.apache.poi.hwpf.usermodel.Range;
import collaborative.common.Pair;

public class Client {
protected Socket sock = null;
protected BufferedReader in = null;
protected PrintWriter out = null;
protected JClientFrame frame = null;
protected Properties prop = null;
protected Properties langProp = null;
protected HashMap<String,String> languagePairMap = null;
protected ArrayList<Sentence> sentenceStates = new ArrayList<Sentence>();
protected Logger logger = Logger.getLogger("client.logging");
protected String userType = null;
protected String language = null;
protected String partnerLanguage = null;
protected int protocolNumber = 0;
protected int sentenceId = 0;
protected int maxModifyCount = 8;
protected File file = null;
protected Pair<Integer, Integer> lastPartnerAction = null;
protected final String WORD_FILE_EXTENSION = ".doc";
protected final String LINE_FEED = "<<linefeed>>";

public Client(){
this.setPropertyInstances();
this.frame = new JClientFrame(this);
this.logger.log(Level.INFO, "Start Client!");
}

public void createSocket() throws ConnectException, IOException{
String host = this.getProperty("server.host");
int port = Integer.parseInt(this.getProperty("server.port"));
this.sock = new Socket(host, port);
this.in = new BufferedReader(new InputStreamReader(this.sock.getInputStream()));
this.out = new PrintWriter(this.sock.getOutputStream(), true);
}

public void initiate(String lang, String pLang, int protocol, File file) throws IOException{
this.createSocket();
this.userType = "source";
this.language = lang;

```

```

this.partnerLanguage = pLang;
this.protocolNumber = protocol;
this.file = file;
this.send(this.userType+" "+this.protocolNumber+" "+this.language+" "+this.partnerLanguage);
logger.setLevel(Level.ALL);
new Reader(this);
}

public void initiate(String language, int protocolNumber) throws IOException{
this.createSocket();
this.userType = "target";
this.language = language;
this.protocolNumber = protocolNumber;
this.send(this.userType+" "+this.protocolNumber+" "+this.language);
logger.setLevel(Level.ALL);
new Reader(this);
}

public boolean isSource(){return this.getUserType().equals("source");}
public boolean isTarget(){return this.getUserType().equals("target");}

public boolean isWaiting(){
if(this.getIdOfEditableSentence() == -1) return true;
return false;
}

public boolean isFailed(int id){
return !this.sentenceStates.get(id).isUnderLimitedCount(this.maxModifyCount);
}

public JProcessView getProcessView(){return this.frame.mainPerspective.processView;}
public JMessageView getMessageView(){return this.frame.mainPerspective.messageView;}
public JStateView getStatePane(){return this.frame.mainPerspective.statePane;}
public Logger getLogger(){return this.logger;}
public Container getContentPane(){return this.frame.getContentPane();}
public String getProperty(String property){return this.prop.getProperty(property);}
public int getProtocolNumber(){return this.protocolNumber;}
public String getUserType(){return this.userType;}
public int getId(){return this.sentenceId;}
public int getWorkCode(int id){return this.sentenceStates.get(id).getWorkCode();}
public int getModifyCount(int id){return this.sentenceStates.get(id).getModifyCount();}
public void getOriginal(int number){this.send(110, number, "");}
public void getModifiedOriginal(int number){this.send(111, number, "");}
public void getTranslation(int number){this.send(112, number, "");}
public void getModifiedTranslation(int number){this.send(113, number, "");}
public void getBackTranslation(int number){this.send(114, number, "");}
public void getReBackTranslation(int number){this.send(115, number, "");}
public void getModifiedTranslationForConfirmation(int number){this.send(116, number, "");}
public void getBackTranslationInModifying(int number){this.send(117, number, "");}
public void getReBackTranslationInModifying(int number){this.send(118, number, "");}
public void getModifiedTranslationFirst(int number){this.send(119, number, "");}
public void getModifiedTranslationNotFirst(int number){this.send(120, number, "");}
public void getInitTasks(){this.send(104,0,"");}

```



```

public String getFileContent(){
String ret = null;
try{
if(this.file.getName().toLowerCase().endsWith(this.WORD_FILE_EXTENSION)){
HWPFDocument wordFile = new HWPFDocument(new FileInputStream(file));
Range range = wordFile.getRange();
for(int i=0;i<range.numParagraphs();i++) ret+=range.getParagraph(i);
}
} catch(IOException e){e.printStackTrace();}
return ret;
}

public String getLanguageProperty(String property){return this.langProp.getProperty(property);}

public String getLanguageProperty(String property, Object replace){
String[] temp = {replace.toString()};
return this.getLanguageProperty(property, temp);
}

public String getLanguageProperty(String property, String[] replace){
ArrayList<String> replaceList = new ArrayList<String>(Arrays.asList(replace));
String mes = this.langProp.getProperty(property);
for(int i=0;i<replaceList.size();i++){
mes = mes.replaceAll("\\\\"+(i+1)+"\\*", replaceList.get(i));
}
return mes;
}

public int getIdOfEditableSentence(){
for(int i=0;i<this.sentenceStates.size();i++)
if(this.isSource())
if(this.sentenceStates.get(i).getWorkCode()-100*this.getProtocolNumber() < 10)return i;
else if(this.isTarget())
if(this.sentenceStates.get(i).getWorkCode()-100*this.getProtocolNumber() < 20 &&
this.sentenceStates.get(i).getWorkCode()-100*this.getProtocolNumber() >= 10)return i;
return -1;
}

public HashMap<String,String> getLanguagePairMap(){
if(this.languagePairMap == null){
this.languagePairMap = new HashMap<String,String>();
String supported = this.getProperty("language.pair");
ArrayList<String> supportedPair = new ArrayList<String>(Arrays.asList(supported.split(";")));
for(String s : supportedPair){
String[] keyval = s.split(":");
this.languagePairMap.put(keyval[0], keyval[1]);
}
}
return this.languagePairMap;
}

public void setUserType(String type){this.userType = type;}

```

```

public void setFile(File file){this.file = file;}
public void setId(int id){this.sentenceId = id;}
public void setPerspective(String perspective){this.frame.setPerspective(perspective);}
public void setOriginal(int number, String text){this.send(210, number, text);}
public void setModifiedOriginal(int number, String text){this.send(211, number, text);}
public void setTranslation(int number, String text){this.send(212, number, text);}
public void setModifiedTranslation(int number, String text){this.send(213, number, text);}
public void setBackTranslation(int number, String text){this.send(214, number, text);}
public void setReBackTranslation(int number, String text){this.send(215, number, text);}
public void setWorkCode(int id, int code){this.sentenceStates.get(id).setWorkCode(code);}

public void setPropertyInstances(){
this.prop = new Properties();
try{this.prop.loadFromXML(new FileInputStream("Colab_client.properties"));
} catch(IOException e) {e.printStackTrace();}
this.langProp = new Properties();
try{this.langProp.loadFromXML(new FileInputStream("lang/lang_ja.properties"));
} catch(IOException e) {e.printStackTrace();}
}

public void setInitWorkCodes(String mes){
String[] codes = mes.split(",");
for(int i=0;i<codes.length;i++) this.sentenceStates.add(new Sentence(Integer.parseInt(codes[i])));
}

public void setLastPartnerAction(int id, int workCode){
this.lastPartnerAction = new Pair<Integer, Integer>(id, workCode);
}

public void sendWorkCode(int id, int code){
this.setWorkCode(id, code);
this.send(203,id,Integer.toString(code));
}

public void incrementModifyCount(int id){this.sentenceStates.get(id).incrementCount();}
public void resetModifyCount(int id){this.sentenceStates.get(id).resetCount();}

public void sendAllSentence(){
String ret = "";
try{
if(this.file.getName().toLowerCase().endsWith(this.WORD_FILE_EXTENSION)){
HWPFDocument wordFile = new HWPFDocument(new FileInputStream(file));
Range range = wordFile.getRange();
for(int i=0;i<range.numParagraphs();i++){
if(range.getParagraph(i).text() == null) continue;
ret+=range.getParagraph(i).text();
}
}
ret = ret.replaceAll("((\\r)?\\n)|(\\r(\\n)?)", this.LINE_FEED);
this.send(200,0,ret);
} catch(IOException e){e.printStackTrace();}
}

```

```

public void send(String message){this.out.println(message);}

public void send(int code, int sentenceNumber, String message){
String numberStr = String.valueOf(sentenceNumber);
for(int i=numberStr.length();i<4;i++)numberStr = "0"+numberStr;
this.send(code+numberStr+message);
this.getLogger().log(Level.INFO, "Send following message to Server: "+message);
}

public void showFrame(){this.frame.showFrame();}

public boolean next(){
int nextId = this.getIdOfEditableSentence();
if(nextId == -1) {
if(this.finished()) this.getProcessView().setFinishedView();
else if(!this.getProcessView().isWaitingView()) this.getProcessView().setWaitingView();
else this.getProcessView().notifyPartnerAction(this.lastPartnerAction);
return false;
}
this.frame.mainPerspective.hideProcessView(nextId);
this.getProcessView().setView(this.sentenceStates.get(nextId).getWorkCode(), nextId);
Thread tmp = new Thread(){
public void run(){
try {Thread.sleep(600);}
catch (InterruptedException e) {}
frame.mainPerspective.showProcessView();
}
};
tmp.start();
return true;
}

public boolean finished(){
for(Sentence c : this.sentenceStates) if(c.getWorkCode()-100*this.getProtocolNumber() < 20) return false;
return true;
}

public static void main(String[] args) {
Client core = new Client();
core.userType = "source";
core.setPerspective("config");
core.showFrame();
}
}

```

### **A.2.9 Reader.java**

```

package collaborative.client;
import java.io.BufferedReader;
import java.io.IOException;
import java.net.SocketException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.regex.Matcher;

```

```

import java.util.regex.Pattern;
import javax.swing.JOptionPane;

public class Reader extends Thread {
protected Client core = null;
protected BufferedReader in = null;

public Reader(Client core){
super();
this.core = core;
this.in = this.core.in;
this.start();
}

public void setOriginal(String message){this.core.getProcessView().setOriginal(
Integer.parseInt(message.substring(0, 4)), message.substring(4));}
public void setModifiedOriginal(String message){this.core.getProcessView().setModifiedOriginal(
Integer.parseInt(message.substring(0, 4)), message.substring(4));}
public void setTranslation(String message){this.core.getProcessView().setTranslation(
Integer.parseInt(message.substring(0, 4)), message.substring(4));}
public void setModifiedTranslation(String message){this.core.getProcessView().
setModifiedTranslation(Integer.parseInt(message.substring(0, 4)), message.substring(4));}
public void setModifiedTranslationFirst(String message){this.core.getProcessView().
setModifiedTranslationFirst(Integer.parseInt(message.substring(0, 4)), message.substring(4));}
public void setModifiedTranslationNotFirst(String message){this.core.getProcessView().
setModifiedTranslationNotFirst(Integer.parseInt(message.substring(0, 4)), message.substring(4));}
public void setModifiedTranslationForConfirmation(String message){this.core.getProcessView().
setModifiedTranslationForConfirmation(Integer.parseInt(message.substring(0, 4)), message.substring(4));}
public void setBackTranslation(String message){this.core.getProcessView().setBackTranslation(
Integer.parseInt(message.substring(0, 4)), message.substring(4));}
public void setBackTranslationInModifying(String message){this.core.getProcessView().
setBackTranslationInModifying(Integer.parseInt(message.substring(0, 4)), message.substring(4));}
public void setReBackTranslation(String message){this.core.getProcessView().setReBackTranslation(
Integer.parseInt(message.substring(0, 4)), message.substring(4));}
public void setReBackTranslationInModifying(String message){this.core.getProcessView().
setReBackTranslationInModifying(Integer.parseInt(message.substring(0, 4)), message.substring(4));}

public ArrayList<String> getArrayFromRegex(String message, String regex){
ArrayList<String> temp = new ArrayList<String>();
Matcher m = Pattern.compile(regex).matcher(message);
while(m.find())temp.add(message.substring(m.start(), m.end()));
return temp;
}

public void onConnection(int maxCycle){
this.core.setPerspective("main");
this.core.maxModifyCount = maxCycle;
if(this.core.isSource()) this.core.sendAllSentence();
}

public boolean decode(String message){
this.core.getLogger().log(Level.INFO, "Receive following message from Server: "+message);
if(message.matches("[0-9]{7}.*")){int code = Integer.parseInt(message.substring(0, 3));}
}

```

```

switch(code){
case 203:
this.core.setLastPartnerAction(Integer.parseInt(message.substring(3, 7)),
Integer.parseInt(message.substring(7)));
if(this.core.isWaiting()){
this.core.setWorkCode(Integer.parseInt(message.substring(3, 7)), Integer.parseInt(message.substring(7)));
this.core.next();
} else this.core.setWorkCode(Integer.parseInt(message.substring(3, 7)),
Integer.parseInt(message.substring(7)));
break;
case 205:this.core.setInitWorkCodes(message.substring(7));
this.core.getStatePane().setNumberOfAllSentences(this.core.sentenceStates.size());
this.core.next();break;
case 210:this.setOriginal(message.substring(3));break;
case 211:this.setModifiedOriginal(message.substring(3));break;
case 212:this.setTranslation(message.substring(3));break;
case 213:this.setModifiedTranslation(message.substring(3));break;
case 214:this.setBackTranslation(message.substring(3));break;
case 215:this.setReBackTranslation(message.substring(3));break;
case 216:this.setModifiedTranslationForConfirmation(message.substring(3));break;
case 217:this.setBackTranslationInModifying(message.substring(3));break;
case 218:this.setReBackTranslationInModifying(message.substring(3));break;
case 219:this.setModifiedTranslationFirst(message.substring(3));break;
case 220:this.setModifiedTranslationNotFirst(message.substring(3));break;
case 900:return true;
case 901: this.onConnection(Integer.parseInt(message.substring(7)));break;
}
}
return false;
}

public void run(){
while(true)
try{
String mes = this.in.readLine();
if(this.decode(mes)){
this.core.getLogger().log(Level.INFO, "Partner's client shutdown ");
String[] obj = {this.core.getLanguageProperty("perspective.main.partner.exit.1"),
this.core.getLanguageProperty("perspective.main.partner.exit.2")};
JOptionPane.showConfirmDialog(this.core.frame, obj,
this.core.getLanguageProperty("perspective.main.partner.exit.title"),
JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
break;
}
}catch(SocketException e){
try {this.in.close();this.core.out.close();this.core.sock.close();}
catch (IOException e1) {e1.printStackTrace();}System.exit(0);
} catch(IOException e){e.printStackTrace();}
}
}

```

## A.2.10 Protocol.java

```
package collaborative.client;
```

```

import java.awt.Color;
import java.awt.event.MouseEvent;
import java.util.ArrayList;
import java.util.Arrays;
import javax.swing.JRadioButton;
import javax.swing.JTextArea;

public class Protocol {
public static void setProcessView(int code, int id, final JProcessView pane){
Integer[] nextCodes = null;
Boolean[] areNexts = null;
switch(code){
case 100:
setModifyView(pane, getModifyAction(pane, code, 111), getSourceModifyLabels(pane), id);
pane.getOriginal(id);
pane.getBackTranslationInModifying(id);
pane.getModifiedOriginal(id);
break;
case 101:
nextCodes = (Integer[])Arrays.asList(new Integer(120), new Integer(100)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(false)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, true, null),
getMouseOnRadioAction(pane, code, 2), getSourceConfirmLabels(pane), getRadioMessage(pane, BI_ACCURACY_SOURCE), id);
pane.getOriginal(id);
pane.getBackTranslation(id);
break;
case 110:
setModifyView(pane, getModifyAction(pane, code, 101), getTargetModifyLabels(pane), id);
pane.getTranslation(id);
pane.getModifiedTranslation(id);
break;
case 111:
nextCodes = (Integer[])Arrays.asList(new Integer(101), new Integer(110)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(false)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, false, null),
getMouseOnRadioAction(pane, code, 2), getTargetFluencyConfirmLabels(pane), getRadioMessage(pane, BI_FLUENCY), id);
pane.getTranslation(id);
break;
case 200:
setModifyView(pane, getModifyAction(pane, code, 201), getSourceModifyLabels(pane), id);
pane.getOriginal(id);
pane.getBackTranslationInModifying(id);
pane.getModifiedOriginal(id);
break;
case 201:
nextCodes = (Integer[])Arrays.asList(new Integer(211), new Integer(200)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(false)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, true, null),
getMouseOnRadioAction(pane, code, 2), getSourceConfirmLabels(pane), getRadioMessage(pane, BI_ACCURACY_SOURCE), id);
pane.getOriginal(id);
pane.getBackTranslation(id);
break;
case 202:

```

```

nextCodes = (Integer[])Arrays.asList(new Integer(220), new Integer(213)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(true)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, false, null),
getMouseOnRadioAction(pane, code, 2), getSourceConfirmLabels(pane), getRadioMessage(pane, BI_ACCURACY_SOURCE), id);
pane.getOriginal(id);
pane.getBackTranslation(id);
break;
case 210:
if(pane.getCore().isFailed(id)){
pane.getCore().sendWorkCode(id, 221);
pane.getCore().next();
} else {
setModifyView(pane, getModifyAction(pane, code, 212), getTargetModifyLabels(pane), id);
if(pane.getCore().getModifyCount(id) == 0) pane.getModifiedTranslationFirst(id);
else {
pane.getReBackTranslationInModifying(id);
pane.getModifiedTranslationNotFirst(id);
}
pane.getTranslation(id);
}
break;
case 211:
nextCodes = (Integer[])Arrays.asList(new Integer(220), new Integer(210)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(false)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, false, null),
getMouseOnRadioAction(pane, code, 2), getTargetFluencyConfirmLabels(pane), getRadioMessage(pane, BI_FLUENCY), id);
pane.getTranslation(id);
break;
case 212:
nextCodes = (Integer[])Arrays.asList(new Integer(202), new Integer(210)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(false)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, false, null),
getMouseOnRadioAction(pane, code, 2), getTargetAccuracyConfirmLabels(pane),
getRadioMessage(pane, BI_ACCURACY_TARGET), id);
pane.getModifiedTranslationForConfirmation(id);
pane.getReBackTranslation(id);
break;
case 213:
if(pane.getCore().isFailed(id)){
pane.getCore().sendWorkCode(id, 221);
pane.getCore().next();
} else {
setModifyView(pane, getModifyAction(pane, code, 212), getTargetModifyLabels(pane), id);
pane.getTranslation(id);
pane.getReBackTranslationInModifying(id);
pane.getModifiedTranslationNotFirst(id);
}
break;
case 300:
setModifyView(pane, getModifyAction(pane, code, 303), getSourceModifyLabels(pane), id);
pane.getOriginal(id);
pane.getBackTranslationInModifying(id);
pane.getModifiedOriginal(id);

```

```

break;
case 301:
setModifyView(pane, getModifyAction(pane, code, 303), getSourceModifyLabels(pane), id);
pane.getOriginal(id);
pane.getBackTranslationInModifying(id);
pane.getModifiedOriginal(id);
break;
case 302: // プロトコル 3 原文修正 翻訳文側で失敗
if(pane.getCore().isFailed(id)){
pane.getCore().sendWorkCode(id, 321);
pane.getCore().next();
} else {
setModifyView(pane, getModifyAction(pane, code, 303), getSourceModifyLabels(pane), id);
pane.getOriginal(id);
pane.getBackTranslationInModifying(id);
pane.getModifiedOriginal(id);
}
break;
case 303:
nextCodes = (Integer[])Arrays.asList(new Integer(313), new Integer(314), new Integer(300)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(true), new Boolean(false)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, true, null),
getMouseOnRadioAction(pane, code, 3), getSourceConfirmLabels(pane), getRadioMessage(pane, TRI_ACCURACY), id);
pane.getOriginal(id);
pane.getBackTranslation(id);
break;
case 304:
nextCodes = (Integer[])Arrays.asList(new Integer(320), new Integer(301)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(false)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, true, null),
getMouseOnRadioAction(pane, code, 2), getSourceConfirmLabels(pane),
getRadioMessage(pane, TRI_ACCURACY_MIDDLE_LOWER), id);
pane.getOriginal(id);
pane.getBackTranslation(id);
break;
case 305:
nextCodes = (Integer[])Arrays.asList(new Integer(320), new Integer(311)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(true)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, true, null),
getMouseOnRadioAction(pane, code, 2), getSourceConfirmLabels(pane),
getRadioMessage(pane, TRI_ACCURACY_MIDDLE_LOWER), id);
pane.getOriginal(id);
pane.getBackTranslation(id);
break;
case 310:
setModifyView(pane, getModifyAction(pane, code, 315), getTargetModifyLabels(pane), id);
if(pane.getCore().getModifyCount(id) == 0) pane.getModifiedTranslationFirst(id);
else {
pane.getReBackTranslationInModifying(id);
pane.getModifiedTranslationNotFirst(id);
}
pane.getTranslation(id);
break;

```



```

case 311:
if(pane.getCore().isFailed(id)){
pane.getCore().sendWorkCode(id, 321);
pane.getCore().next();
} else {
setModifyView(pane, getModifyAction(pane, code, 315), getTargetModifyLabels(pane), id);
pane.getTranslation(id);
pane.getReBackTranslationInModifying(id);
pane.getModifiedTranslationNotFirst(id);
}
break;
case 312:
if(pane.getCore().getModifyCount(id) == 0)
setModifyView(pane, getModifyAction(pane, code, 316), getTargetModifyLabels(pane), id);
if(pane.getCore().getModifyCount(id) == 0) pane.getModifiedTranslationFirst(id);
else {
pane.getReBackTranslationInModifying(id);
pane.getModifiedTranslationNotFirst(id);
}
pane.getTranslation(id);
break;
case 313:
nextCodes = (Integer[])Arrays.asList(new Integer(320), new Integer(310), new Integer(321)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(false), new Boolean(true)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, false, null),
getMouseOnRadioAction(pane, code, 3), getTargetFluencyConfirmLabels(pane), getRadioMessage(pane, TRI_FLUENCY), id);
pane.getTranslation(id);
break;
case 314:
nextCodes = (Integer[])Arrays.asList(new Integer(302), new Integer(312), new Integer(302)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(false), new Boolean(true)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, true, 302),
getMouseOnRadioAction(pane, code, 3), getTargetFluencyConfirmLabels(pane), getRadioMessage(pane, TRI_FLUENCY), id);
pane.getTranslation(id);
break;
case 315:
nextCodes = (Integer[])Arrays.asList(new Integer(305), new Integer(310)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(false)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, true, null),
getMouseOnRadioAction(pane, code, 2), getTargetAccuracyConfirmLabels(pane),
getRadioMessage(pane, TRI_ACCURACY_UPPER_MIDDLE), id);
pane.getModifiedTranslationForConfirmation(id);
pane.getReBackTranslation(id);
break;
case 316:
nextCodes = (Integer[])Arrays.asList(new Integer(304), new Integer(312)).toArray();
areNexts = (Boolean[])Arrays.asList(new Boolean(true), new Boolean(false)).toArray();
setConfigView(pane, getConfirmAction(pane, code, nextCodes, areNexts, true, 302),
getMouseOnRadioAction(pane, code, 2), getTargetAccuracyConfirmLabels(pane),
getRadioMessage(pane, TRI_ACCURACY_UPPER_MIDDLE), id);
pane.getModifiedTranslationForConfirmation(id);
pane.getReBackTranslation(id);
break;

```

```

}
setDefaultMessage(pane,code,true);
pane.getCore().getMessageView().showDefaultText();
}

private static void setModifyView(JProcessView pane, OnMouseAction onButtonAction,
String[] labels, int id){
pane.setValidTextAreas(labels.length, labels);
pane.setTextAreaEditable(labels.length-1, true);
for(int i=labels.length-2;i>=0;i--) pane.setTextAreaEditable(i, false);
pane.setValidRadioButtons(0);
pane.getButtonGroup().clearSelection();
pane.setButtonVisible(true,pane.getCore().getLanguageProperty("perspective.main.process.button.decision"));
pane.setButtonMouseAction(onButtonAction);
pane.setMessageVisible(false);
JStateView statePane = pane.getCore().getStatePane();
statePane.setModifyCount(pane.getCore().getModifyCount(id)+1);
statePane.setSentenceNumber(id+1);
}

private static void setConfigView(JProcessView pane, OnMouseAction onButtonAction,
OnMouseAction[] onRadioAction, String[] textAreaLabels, String[] configLabels, int id){
pane.setValidTextAreas(textAreaLabels.length, textAreaLabels);
pane.setTextAreaEditable(0, false);
pane.setTextAreaEditable(1, false);
pane.setValidRadioButtons(configLabels.length, configLabels);
pane.setRadioButtonsMouseAction(onRadioAction);
pane.getButtonGroup().clearSelection();
pane.setButtonVisible(true,pane.getCore().getLanguageProperty("perspective.main.process.button.decision"));
pane.setButtonMouseAction(onButtonAction);
pane.setMessageVisible(false);
JStateView statePane = pane.getCore().getStatePane();
statePane.setModifyCount(pane.getCore().getModifyCount(id)+1);
statePane.setSentenceNumber(id+1);
}

public static void setWaitingView(JProcessView pane){
pane.setValidTextAreas(0);
pane.setValidRadioButtons(0);
pane.setButtonVisible(false);
pane.setMessageVisible(true,pane.getCore().getLanguageProperty("perspective.main.process.state.waiting"));
pane.setDefaultMessage(pane.getCore().getLanguageProperty("message.waiting"));
pane.getCore().getMessageView().showDefaultText();
pane.getCore().getStatePane().setNoSentencesWorked();
}

public static void setFinishView(JProcessView pane){
pane.setValidTextAreas(0);
pane.setValidRadioButtons(0);
pane.setButtonVisible(false);
pane.setMessageVisible(true,pane.getCore().getLanguageProperty("perspective.main.process.state.accomplish"));
pane.setDefaultMessage(pane.getCore().getLanguageProperty("message.accomplish"));
pane.getCore().getMessageView().showDefaultText();
}

```

```

pane.getCore().getStatePane().setNoSentencesWorked();
}

private static OnMouseAction getModifyAction(final JProcessView pane,
final int code, final int nextCode){
return new OnMouseAction(pane, pane.getCore().getLanguageProperty("message."+code+".decision")){
public void mouseClicked(MouseEvent arg0){
Client core = pane.getCore();
JTextArea area = null;
if(core.isSource()) area = pane.getModifiedOriginalTextArea();
else if(core.getProtocolNumber() == 1) area = pane.getModifiedTranslationTextArea();
else if(core.getModifyCount(core.getId()) == 0) area = pane.getModifiedTranslationFirstTextArea();
else area = pane.getModifiedTranslationNotFirstTextArea();
if(area.getText().matches("(\\s| )*")){
pane.setMessageVisible(true,
core.getLanguageProperty("perspective.main.process.error.modify_blank"), Color.RED);
} else {
if(core.isSource()) core.setModifiedOriginal(pane.getId(), area.getText());
else core.setModifiedTranslation(pane.getId(), area.getText());
core.sendWorkCode(pane.getId(), nextCode);core.incrementModifyCount(pane.getId());
if(core.getProtocolNumber() == 1) core.next();
else Protocol.setProcessView(nextCode, pane.getId(), pane);
}
}
};
}

private static OnMouseAction getConfirmAction(final JProcessView pane, final int code,
final Integer[] nextCodes, final Boolean[] areNexts, final boolean checkFinish, final Integer finishCode){
return new OnMouseAction(pane){
public void mouseEntered(MouseEvent arg0) {
this.messageView.showText(pane.getCore().getLanguageProperty("message."+code+".decision."+
(this.processView.getSelectedRadioButtonNumber()+1)));
}

public void mouseClicked(MouseEvent arg0){
Client core = pane.getCore();
if(pane.getButtonGroup().getSelection() == null){
pane.setMessageVisible(true, core.getLanguageProperty("perspective.main.process.error.not_selected"),
Color.RED);
} else {
JRadioButton[] radios = pane.getRadioButtons();
for(int i=0;i<nextCodes.length;i++){
if(radios[i].isSelected()){
if(checkFinish && i == nextCodes.length-1){
if(core.isFailed(pane.getId())){
if(finishCode == null){
core.sendWorkCode(pane.getId(), 21+core.getProtocolNumber()*100);
core.next();
} else {
core.sendWorkCode(pane.getId(), finishCode);
core.resetModifyCount(pane.getId());
core.next();
}
}
}
}
}
}
}
}
}

```

```

break;
}
}
core.sendWorkCode(pane.getId(), nextCodes[i]);
if(areNexts[i]) core.next(); else Protocol.setProcessView(nextCodes[i], pane.getId(), pane);
break;
}
}
}
}
};
}

private static String[] getTargetModifyLabels(JProcessView pane){
if(pane.getCore().getProtocolNumber() == 1 || pane.getCore().getModifyCount(pane.getId()) == 0)
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.translation")),
new String(pane.getCore().getLanguageProperty("perspective.main.process.modify")))
.toArray();
else
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.translation")),
new String(pane.getCore().getLanguageProperty("perspective.main.process.back_translation")),
new String(pane.getCore().getLanguageProperty("perspective.main.process.modify")))
.toArray();
}

private static String[] getSourceModifyLabels(JProcessView pane){
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.original")),
new String(pane.getCore().getLanguageProperty("perspective.main.process.back_translation")),
new String(pane.getCore().getLanguageProperty("perspective.main.process.modify")))
.toArray();
}

private static String[] getSourceConfirmLabels(JProcessView pane){
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.original")),
new String(pane.getCore().getLanguageProperty("perspective.main.process.back_translation")))
.toArray();
}

private static String[] getTargetAccuracyConfirmLabels(JProcessView pane){
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.modified_translation")),
new String(
pane.getCore().getLanguageProperty("perspective.main.process.modified_back_translation")))
.toArray();
}

private static String[] getTargetFluencyConfirmLabels(JProcessView pane){
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.translation")))
.toArray();
}

private static OnMouseAction[] getMouseOnRadioAction(JProcessView pane, int code, int num){
ArrayList<OnMouseAction> list = new ArrayList<OnMouseAction>();
for(int i=1;i<=num;i++) list.add(

```

```

new OnMouseAction(pane, pane.getCore().getLanguageProperty("message."+code+".radio."+i));
return (OnMouseAction[])list.toArray();
}

private static void setDefaultMessage(JProcessView pane, int code, boolean checkFirst){
if(checkFirst){
if(pane.getCore().getModifyCount(pane.getId()) == 0)
pane.setDefaultMessage(pane.getCore().getLanguageProperty("message."+code+".default.first"));
else pane.setDefaultMessage(pane.getCore().getLanguageProperty("message."+code+".default.not_first"));
}else pane.setDefaultMessage(pane.getCore().getLanguageProperty("message."+code+".default"));
}

private static String[] getRadioMessage(JProcessView pane, int type){
switch(type){
case BI_ACCURACY_SOURCE:
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.bi.upper",
pane.getCore().getLanguageProperty("perspective.main.process.original"))),
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.bi.lower",
pane.getCore().getLanguageProperty("perspective.main.process.original")))).toArray();
case BI_ACCURACY_TARGET:
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.bi.upper",
pane.getCore().getLanguageProperty("perspective.main.process.modified_translation"))),
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.bi.lower",
pane.getCore().getLanguageProperty("perspective.main.process.modified_translation")))).toArray();
case BI_FLUENCY:
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.fluency.bi.upper")),
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.fluency.bi.lower"))).toArray();
case TRI_ACCURACY:
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.tri.upper",
pane.getCore().getLanguageProperty("perspective.main.process.original"))),
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.tri.middle",
pane.getCore().getLanguageProperty("perspective.main.process.original"))),
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.tri.lower",
pane.getCore().getLanguageProperty("perspective.main.process.original")))).toArray();
case TRI_ACCURACY_UPPER_MIDDLE:
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.tri.middle_upper",
pane.getCore().getLanguageProperty("perspective.main.process.modified_translation"))),
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.tri.lower",
pane.getCore().getLanguageProperty("perspective.main.process.modified_translation")))).toArray();
case TRI_ACCURACY_MIDDLE_LOWER:
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.tri.upper",
pane.getCore().getLanguageProperty("perspective.main.process.original"))),
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.accuracy.tri.middle_lower",
pane.getCore().getLanguageProperty("perspective.main.process.original")))).toArray();
case TRI_FLUENCY:
return (String[])Arrays.asList(
new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.fluency.tri.upper")),

```

```

new String(pane.getCore().getLanguageProperty("perspective.main.process.radio.fluency.tri.middle")),
new String(
pane.getCore().getLanguageProperty("perspective.main.process.radio.fluency.tri.lower")))
.toArray();
}
return null;
}

private final static int BI_ACCURACY_SOURCE = 210;
private final static int BI_ACCURACY_TARGET = 211;
private final static int BI_FLUENCY = 22;
private final static int TRI_ACCURACY = 310;
private final static int TRI_ACCURACY_UPPER_MIDDLE = 311;
private final static int TRI_ACCURACY_MIDDLE_LOWER = 312;
private final static int TRI_FLUENCY = 32;
}

```

### A.2.11 JClientFrame.java

```

package collaborative.client;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.util.logging.Level;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

public class JClientFrame extends JFrame{
protected Client core = null;
protected JSettingPerspective configPerspective = null;
protected JMainPerspective mainPerspective = null;
protected JWaitingPerspective waitingPerspective = null;
protected JSubMenu submenu = null;
private static final long serialVersionUID = -6208985002371685202L;

public JClientFrame(final Client core){
super();
this.core = core;
this.submenu = new JSubMenu(this.core, this);
this.setJMenuBar(this.submenu);
final JClientFrame frame = this;
this.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e){
String obj = core.getLanguageProperty("perspective.main.frame.exit_confirm.1");
if(JOptionPane.showConfirmDialog(frame, obj,
core.getLanguageProperty("perspective.main.frame.exit_confirm.title"),
JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE) == JOptionPane.YES_OPTION) System.exit(0);
}
});
this.setTitle(this.core.getLanguageProperty("frame.title"));
this.setResizable(false);
}

public void setPerspective(String perspective){

```

```

this.core.getLogger().log(Level.INFO, "Change Perspective to: \""+perspective+"\" perspective");
if(perspective.equals("main")){
if(this.waitingPerspective != null) this.waitingPerspective.setVisible(false);
if(this.configPerspective != null) this.configPerspective.setVisible(false);
if(this.mainPerspective == null) {
this.mainPerspective = new JMainPerspective(this, this.core);
this.getContentPane().add(this.mainPerspective);
}
this.mainPerspective.setVisible(true);
} else if(perspective.equals("config")){
if(this.waitingPerspective != null) this.waitingPerspective.setVisible(false);
if(this.mainPerspective != null) this.mainPerspective.setVisible(false);
if(this.configPerspective == null) {
this.configPerspective = new JSettingPerspective(this, this.core);
this.getContentPane().add(this.configPerspective);
}
this.configPerspective.setVisible(true);
} else if(perspective.equals("waiting")){
if(this.configPerspective != null) this.configPerspective.setVisible(false);
if(this.mainPerspective != null) this.mainPerspective.setVisible(false);
if(this.waitingPerspective == null) {
this.waitingPerspective = new JWaitingPerspective(this, this.core);
this.getContentPane().add(this.waitingPerspective);
}
this.waitingPerspective.setVisible(true);
}
this.getContentPane().validate();
}

public void showFrame(){
this.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
String[] size = this.core.getProperty("frame.size").split(",");
int width = Integer.parseInt(size[0]);
int height = Integer.parseInt(size[1]);
this.setBounds((d.width-width)/2,(d.height-height)/2,width,height);
this.setVisible(true);
}
}

```

## A.2.12 JSettingPerspective.java

```

package collaborative.client;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import java.net.ConnectException;

```

```

import java.net.UnknownHostException;
import java.util.HashMap;
import javax.swing.AbstractButton;
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JTextField;
import javax.swing.filechooser.FileFilter;
import collaborative.common.Pair;

public class JSettingPerspective extends JPanel implements ActionListener{
protected Client core = null;
protected JClientFrame parent = null;
protected JLabel fileErrorLabel = null;
protected JComboBox sourceLanguageSelect = null;
protected JComboBox targetLanguageSelect = null;
protected ButtonGroup usertypes = null;
protected JRadioButton sourceRadio = null;
protected JRadioButton targetRadio = null;
protected JComboBox protocolSelect = null;
protected JTextField fileNameField = null;
protected JFileChooser filechooser = null;
protected File file = null;
protected JButton fileSelectButton = null;
protected JButton okButton = null;
protected JButton clearButton = null;
private static final long serialVersionUID = 3630721213877802391L;
public final int LABEL_WIDTH = 100;
public final int LABEL_HEIGHT = 20;
public final int COLUMN_SIZE = 20;

public JSettingPerspective(JClientFrame parent, Client core){
super();
this.parent = parent;
this.core = core;
this.make();
}

public void make(){
GridBagLayout gridbag = new GridBagLayout();
GridBagConstraints c = new GridBagConstraints();
this.setLayout(gridbag);
c.gridwidth = GridBagConstraints.REMAINDER;
c.anchor = GridBagConstraints.WEST;
this.fileErrorLabel = new JLabel();
gridbag.setConstraints(this.fileErrorLabel, c);
this.add(this.fileErrorLabel);
JPanel langPane = this.getLangPane();

```



```

gridbag.setConstraints(langPane, c);
this.add(langPane);
JPanel userTypePane = this.getUserTypePane();
gridbag.setConstraints(userTypePane, c);
this.add(userTypePane);
JPanel protocolPane = this.getProtocolPane();
gridbag.setConstraints(protocolPane, c);
this.add(protocolPane);
JPanel targetPane = this.getTargetPane();
gridbag.setConstraints(targetPane, c);
this.add(targetPane);
JPanel filePane = this.getFilePane();
gridbag.setConstraints(filePane, c);
this.add(filePane);
JPanel confirmPane = this.getConfirmPane();
gridbag.setConstraints(confirmPane, c);
this.add(confirmPane);
}

public boolean isValidInput(){
if(this.sourceRadio.isSelected()) if(!this.fileNameField.getText().matches("\\s*")) return true;
else return true;
return false;
}

public void clear(){
this.sourceLanguageSelect.setSelectedIndex(0);
this.protocolSelect.setSelectedIndex(0);
this.targetLanguageSelect.setSelectedIndex(0);
this.fileNameField.setText("");
this.file = null;
}

public JPanel getLangPane(){
JPanel langPane = new JPanel();
this.sourceLanguageSelect = new JComboBox();
JLabel languageLabel = new JLabel(this.getCore().getLanguageProperty("perspective.config.language")+":" );
languageLabel.setPreferredSize(new Dimension(this.LABEL_WIDTH, this.LABEL_HEIGHT));
langPane.add(languageLabel);
langPane.add(this.sourceLanguageSelect);
this.sourceLanguageSelect.setEditable(true);
((JTextField)this.sourceLanguageSelect.getEditor().getEditorComponent()).setColumns(this.COLUMN_SIZE);
HashMap<String,String> langMap = core.getLanguagePairMap();
for(String s : langMap.keySet())
this.sourceLanguageSelect.addItem(
new ComboBoxPair<String,String>(s, this.getCore().getLanguageProperty("language."+s)));
this.sourceLanguageSelect.setEditable(false);
this.sourceLanguageSelect.addActionListener(this);
return langPane;
}

public JPanel getUserTypePane(){
JPanel userTypePane = new JPanel(new FlowLayout());

```

```

this.usertypes = new ButtonGroup();
JLabel workLabel = new JLabel(this.getCore().getLanguageProperty("perspective.config.work")+":" );
workLabel.setPreferredSize(new Dimension(this.LABEL_WIDTH, this.LABEL_HEIGHT));
userTypePane.add(workLabel);
JPanel buttonPane = new JPanel(new GridLayout(2,1));
this.sourceRadio = new JRadioButton(
this.getCore().getLanguageProperty("perspective.config.work.source"), true);
this.usertypes.add((AbstractButton)buttonPane.add(this.sourceRadio));
this.targetRadio = new JRadioButton(
this.getCore().getLanguageProperty("perspective.config.work.target"), false);
this.usertypes.add((AbstractButton)buttonPane.add(this.targetRadio));
this.sourceRadio.addActionListener(this);
this.targetRadio.addActionListener(this);
userTypePane.add(buttonPane);
return userTypePane;
}

public JPanel getProtocolPane(){
JPanel protocolPane = new JPanel();
this.protocolSelect = new JComboBox();
JLabel protocolLabel = new JLabel(this.getCore().getLanguageProperty("perspective.config.protocol")+":" );
protocolLabel.setPreferredSize(new Dimension(this.LABEL_WIDTH, this.LABEL_HEIGHT));
protocolPane.add(protocolLabel);
protocolPane.add(this.protocolSelect);
this.protocolSelect.setEditable(true);
((JTextField)this.protocolSelect.getEditor().getEditorComponent()).setColumns(this.COLUMN_SIZE);
for(String s : this.getCore().getProperty("protocols").split(";")){
this.protocolSelect.addItem(new ComboBoxPair<String,String>(
s, this.getCore().getLanguageProperty("protocol."+s)));
}
this.protocolSelect.setEditable(false);
return protocolPane;
}

public JPanel getTargetPane(){
JPanel targetLangPane = new JPanel();
this.targetLanguageSelect = new JComboBox();
JLabel targetLabel = new JLabel(this.getCore().getLanguageProperty("perspective.config.target")+":" );
targetLabel.setPreferredSize(new Dimension(this.LABEL_WIDTH, this.LABEL_HEIGHT));
targetLangPane.add(targetLabel);
targetLangPane.add(this.targetLanguageSelect);
this.targetLanguageSelect.setEditable(true);
((JTextField)this.targetLanguageSelect.getEditor().getEditorComponent()).setColumns(this.COLUMN_SIZE);
HashMap<String,String> langMap = core.getLanguagePairMap();
for(String s : langMap.get(((ComboBoxPair)this.sourceLanguageSelect.getSelectedItem()).key()).split(","))
this.targetLanguageSelect.addItem(
new ComboBoxPair<String,String>(s, this.getCore().getLanguageProperty("language."+s)));
this.targetLanguageSelect.setEditable(false);
return targetLangPane;
}

public JPanel getFilePane(){
JPanel filePane = new JPanel();

```

```

this.fileNameField = new JTextField(this.COLUMN_SIZE);
this.fileNameField.setEditable(false);
this.fileSelectButton = new JButton(this.getCore().getLanguageProperty("button.file.open"));
JLabel fileLabel = new JLabel(this.getCore().getLanguageProperty("perspective.config.file")+":" );
fileLabel.setPreferredSize(new Dimension(this.LABEL_WIDTH, this.LABEL_HEIGHT));
filePane.add(fileLabel);
filePane.add(this.fileNameField);
filePane.add(this.fileSelectButton);
this.fileSelectButton.addActionListener(this);
this.filechooser = new JFileChooser();
filechooser.addChoosableFileFilter(new FileFilter() {
public boolean accept(File file) {
    if(file.isDirectory()) return true;
    return file.getName().toLowerCase().endsWith(".doc");
}
});
public String getDescription() {return "Microsoft Office Word ファイル (*.doc)";}
});
return filePane;
}

public JPanel getConfirmPane(){
JPanel confirmPane = new JPanel();
this.okButton = new JButton(this.getCore().getLanguageProperty("button.ok"));
this.clearButton = new JButton(this.getCore().getLanguageProperty("button.clear"));
confirmPane.add(okButton);
confirmPane.add(clearButton);
this.okButton.addActionListener(this);
this.clearButton.addActionListener(this);
return confirmPane;
}

public Client getCore(){return this.core;}
public void printError(){
if(this.sourceRadio.isSelected() && this.fileNameField.getText().matches("\\s*")) {
this.fileErrorLabel.setText(this.core.getLanguageProperty("perspective.config.error.file.blank"));
this.fileErrorLabel.setForeground(Color.RED);
}else this.fileErrorLabel.setText("");
}

public void showFileChooser(){
int selected = this.filechooser.showOpenDialog(this);
if (selected == JFileChooser.APPROVE_OPTION){
this.file = this.filechooser.getSelectedFile();
this.fileNameField.setText(this.file.getAbsolutePath());
}
}

public void onOKButtonClicked(){
this.core.setPerspective("waiting");
try{
if(this.sourceRadio.isSelected())
this.core.initiate(
(String)((Pair)this.sourceLanguageSelect.getSelectedItem()).getFirst(),
(String)((Pair)this.targetLanguageSelect.getSelectedItem()).getFirst(),

```

```

Integer.parseInt(((String)((Pair)this.protocolSelect.getSelectedItem()).getFirst()).
substring("protocol".length()), this.file);
else
this.core.initiate(
(String)((Pair)this.sourceLanguageSelect.getSelectedItem()).getFirst(),
Integer.parseInt(((String)
((Pair)this.protocolSelect.getSelectedItem()).getFirst()).substring("protocol".length())));
}catch(UnknownHostException e){this.warn();
}catch(ConnectException e){this.warn();
}catch(IOException e){e.printStackTrace();}
}

public void warn(){
String[] replace = {this.core.getProperty("server.host"), this.core.getProperty("server.port")};
String[] obj = {this.core.getLanguageProperty("perspective.waiting.connect_error.message"),
this.core.getLanguageProperty("perspective.waiting.connect_error.settings", replace),
this.core.getLanguageProperty("perspective.waiting.connect_error.confirm")};
JOptionPane.showConfirmDialog(this.parent, obj, this.core.getLanguageProperty
("perspective.waiting.connect_error.title"), JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
this.core.setPerspective("config");
}

public void actionPerformed(ActionEvent ae){
if(ae.getSource().equals(this.targetRadio))
if(this.targetRadio.isSelected()){
this.targetLanguageSelect.setEnabled(false);
this.fileNameField.setEnabled(false);
this.fileSelectButton.setEnabled(false);
} else {
this.targetLanguageSelect.setEnabled(true);
this.fileNameField.setEnabled(true);
this.fileSelectButton.setEnabled(true);
}
else if(ae.getSource().equals(this.sourceRadio))
if(this.sourceRadio.isSelected()){
this.targetLanguageSelect.setEnabled(true);
this.fileNameField.setEnabled(true);
this.fileSelectButton.setEnabled(true);
} else {
this.targetLanguageSelect.setEnabled(false);
this.fileNameField.setEnabled(false);
this.fileSelectButton.setEnabled(false);
}
else if(ae.getSource().equals(this.sourceLanguageSelect)){
this.targetLanguageSelect.removeAllItems();
HashMap<String,String> langMap = core.getLanguagePairMap();
for(String s : langMap.get(((ComboBoxPair)this.sourceLanguageSelect.getSelectedItem()).key()).split(","))
this.targetLanguageSelect.addItem(
new ComboBoxPair<String,String>(s, this.getCore().getLanguageProperty("language."+s)));
} else if(ae.getSource().equals(this.fileSelectButton))this.showFileChooser();
else if(ae.getSource().equals(this.clearButton))this.clear();
else if(ae.getSource().equals(this.okButton))
if(this.isValidInput())this.onOKButtonClicked();
}

```

```
else this.printError();
}
}
```

### A.2.13 JWaitingPerspective.java

```
package collaborative.client;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class JWaitingPerspective extends JPanel {
    protected Client core = null;
    protected JClientFrame parent = null;
    protected JLabel imageLabel = null;
    protected JLabel messageLabel = null;
    private static final long serialVersionUID = 8241885100724877342L;

    public JWaitingPerspective(JClientFrame parent, Client core){
        super();
        this.parent = parent;
        this.core = core;
        this.make();
    }

    public void make(){
        GridBagLayout gridbag = new GridBagLayout();
        GridBagConstraints c = null;
        this.setLayout(gridbag);
        c = new GridBagConstraints();
        c.gridwidth = GridBagConstraints.REMAINDER;
        this.imageLabel = new JLabel(new ImageIcon("img/ajax_activity_indicators_download_animated_pleasewait.gif"));
        gridbag.setConstraints(this.imageLabel, c);
        this.add(this.imageLabel);
        this.messageLabel = new JLabel(this.core.getLanguageProperty("perspective.waiting.wait"));
        gridbag.setConstraints(this.messageLabel, c);
        this.add(this.messageLabel);
    }
}
```

### A.2.14 JMainPerspective.java

```
package collaborative.client;
import java.awt.Dimension;
import java.awt.GridBagLayout;
import java.awt.GridBagConstraints;
import java.awt.Insets;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class JMainPerspective extends JPanel{
    protected Client core = null;
```

```

protected JClientFrame parent = null;
protected JProcessView processView = null;
protected JMessageView messageView = null;
protected JStateView statePane = null;
protected JPanel waitNextPane = null;
private static final long serialVersionUID = 7096878645928811629L;

public JMainPerspective(JClientFrame parent, Client core){
    super();
    this.parent = parent;
    this.core = core;
    this.make();
}

public void make(){
    GridBagLayout gridbag = new GridBagLayout();
    GridBagConstraints c = null;
    this.setLayout(gridbag);
    this.processView = new JProcessView(this,this.core);
    this.messageView = new JMessageView(this,this.core);
    this.statePane = new JStateView(this,this.core);
    this.waitNextPane = this.getWaitNextPane();
    JPanel processWrapper = new JPanel();
    processWrapper.add(this.processView);
    processWrapper.add(this.waitNextPane);
    c = new GridBagConstraints();
    c.gridheight = GridBagConstraints.REMAINDER;
    c.weightx = 1.0;
    c.weighty = 1.0;
    gridbag.setConstraints(processWrapper, c);
    this.add(processWrapper);
    this.showProcessView();
    c = new GridBagConstraints();
    c.gridwidth = GridBagConstraints.RELATIVE;
    c.weightx = 0.5;
    c.weighty = 0.5;
    gridbag.setConstraints(this.statePane, c);
    this.add(this.statePane);
    c = new GridBagConstraints();
    c.gridwidth = GridBagConstraints.RELATIVE;
    c.gridheight = GridBagConstraints.RELATIVE;
    c.anchor = GridBagConstraints.SOUTH;
    c.insets = new Insets(0,0,40,0);
    c.weightx = 0.5;
    c.weighty = 1.0;
    gridbag.setConstraints(this.messageView, c);
    this.add(this.messageView);
}

public void showProcessView(){
    this.processView.setVisible(true);
    this.waitNextPane.setVisible(false);
    this.processView.getParent().validate();
}

```

```

}

public void hideProcessView(int id){
    this.processView.setVisible(false);
    this.waitNextPane.setVisible(true);
    ((JLabel)this.waitNextPane.getComponent(1)).setText(
        this.core.getLanguageProperty("perspective.main.wait_next", id+1));
    this.processView.getParent().validate();
}

public JPanel getWaitNextPane(){
    JPanel helloPane = new JPanel();
    helloPane.setPreferredSize(new Dimension(this.processView.LABEL_WIDTH,this.processView.LABEL_HEIGHT*5));
    GridBagLayout gridbag = new GridBagLayout();
    GridBagConstraints c = null;
    c = new GridBagConstraints();
    c.gridwidth = GridBagConstraints.REMAINDER;
    helloPane.setLayout(gridbag);
    JLabel imageLabel = new JLabel(new ImageIcon("img/ajax-next-sentence.gif"));
    gridbag.setConstraints(imageLabel, c);
    helloPane.add(imageLabel);
    JLabel messageLabel = new JLabel();
    gridbag.setConstraints(messageLabel, c);
    helloPane.add(messageLabel);
    return helloPane;
}
}

```

### A.2.15 JProcessView.java

```

package collaborative.client;
import java.awt.Color;
import java.awt.Component;
import java.awt.Dimension;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.event.MouseListener;
import javax.swing.ButtonGroup;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JRadioButton;
import collaborative.common.Pair;

public class JProcessView extends JPanel{
    protected Client core = null;
    protected JMainPerspective parent = null;
    protected JLabel[] labels = new JLabel[this.TEXT_AREA_NUMBER];
    protected JTextArea[] areas = new JTextArea[this.TEXT_AREA_NUMBER];
    protected JScrollPane[] textAreaPanes = new JScrollPane[this.TEXT_AREA_NUMBER];
    protected JButton button = null;

```

```

protected JRadioButton[] radios = new JRadioButton[this.RADIOBUTTON_NUMBER];
protected ButtonGroup buttonGroup = null;
protected JLabel generalMessage = null;
protected JPanel[] loadingPanel = new JPanel[this.TEXT_AREA_NUMBER];
private static final long serialVersionUID = -6440551044617734535L;
protected final int TEXT_AREA_WIDTH = 450;
protected final int TEXT_AREA_HEIGHT = 70;
protected final int LABEL_WIDTH = 450;
protected final int LABEL_HEIGHT = 20;
protected final int TEXT_AREA_NUMBER = 3;
protected final int RADIOBUTTON_NUMBER = 3;
protected final Color UNEDITABLE_COLOR = new Color(223,239,255);
protected final Color EDITABLE_COLOR = Color.WHITE;
protected final String LINEFEED_REGEX = "\\[\\linefeed\\]";

public JProcessView(JMainPerspective parent, Client core){
    super();
    this.parent = parent;
    this.core = core;
    this.make();
}

public void make(){
    GridBagLayout gridbag = new GridBagLayout();
    GridBagConstraints c = new GridBagConstraints();
    c.gridwidth = GridBagConstraints.REMAINDER;
    c.anchor = GridBagConstraints.WEST;
    this.setLayout(gridbag);
    for(int i=0;i<this.TEXT_AREA_NUMBER;i++){
        this.areas[i] = new JTextArea();
        this.areas[i].setLineWrap(true);
        this.areas[i].setWrapStyleWord(true);
        this.textAreaPanels[i] = new JScrollPane(this.areas[i]);
        this.textAreaPanels[i].setPreferredSize(new Dimension(this.TEXT_AREA_WIDTH, this.TEXT_AREA_HEIGHT));
        this.labels[i] = new JLabel("");
        this.labels[i].setPreferredSize(new Dimension(this.LABEL_WIDTH,this.LABEL_HEIGHT));
        this.loadingPanel[i] = this.createLoadingPanel();
        gridbag.setConstraints(this.textAreaPanels[i], c);
        gridbag.setConstraints(this.loadingPanel[i], c);
        gridbag.setConstraints(this.labels[i], c);
        this.add(this.labels[i]);
        this.add(this.textAreaPanels[i]);
        this.add(this.loadingPanel[i]);
        this.loadingPanel[i].setVisible(false);
    }
    this.buttonGroup = new ButtonGroup();
    for(int i=0;i<this.RADIOBUTTON_NUMBER;i++){
        this.radios[i] = new JRadioButton();
        gridbag.setConstraints(this.radios[i], c);
        this.buttonGroup.add(this.radios[i]);
        this.add(this.radios[i]);
    }
    this.generalMessage = new JLabel("");
}

```



```

this.generalMessage.setPreferredSize(new Dimension(this.LABEL_WIDTH,this.LABEL_HEIGHT));
gridbag.setConstraints(this.generalMessage, c);
this.add(this.generalMessage);
this.button = new JButton();
gridbag.setConstraints(this.button, c);
this.add(this.button);
}

public void setTranslation(int number, String mes){
if(number == this.core.getId()){
this.getTranslationTextArea().setText(mes);
this.showTextArea(this.getTranslationTextArea());
}
}

public void setOriginal(int number, String mes){
if(number == this.core.getId()){
this.getOriginalTextArea().setText(mes);
this.showTextArea(this.getOriginalTextArea());
}
}

public void setModifiedOriginal(int number, String mes){
if(number == this.core.getId()){
this.getModifiedOriginalTextArea().setText(mes);
this.showTextArea(this.getModifiedOriginalTextArea());
}
}

public void setModifiedTranslation(int number, String mes){
if(number == this.core.getId()){
this.getModifiedTranslationTextArea().setText(mes);
this.showTextArea(this.getModifiedTranslationTextArea());
}
}

public void setModifiedTranslationFirst(int number, String mes){
if(number == this.core.getId()){
this.getModifiedTranslationFirstTextArea().setText(mes);
this.showTextArea(this.getModifiedTranslationFirstTextArea());
}
}

public void setModifiedTranslationNotFirst(int number, String mes){
if(number == this.core.getId()){
this.getModifiedTranslationNotFirstTextArea().setText(mes);
this.showTextArea(this.getModifiedTranslationNotFirstTextArea());
}
}

public void setModifiedTranslationForConfirmation(int number, String mes){
if(number == this.core.getId()){
this.getModifiedTranslationForConfirmationTextArea().setText(mes);
this.showTextArea(this.getModifiedTranslationForConfirmationTextArea());
}
}

public void setBackTranslation(int number, String mes){
if(number == this.core.getId()){
this.getBackTranslationTextArea().setText(mes);
}
}

```

```

this.showTextArea(this.getBackTranslationTextArea());
}
}
public void setBackTranslationInModifying(int number, String mes){
if(number == this.core.getId()){
this.getBackTranslationInModifyingTextArea().setText(mes);
this.showTextArea(this.getBackTranslationInModifyingTextArea());
}
}
public void setReBackTranslation(int number, String mes){
if(number == this.core.getId()){
this.getReBackTranslationTextArea().setText(mes);
this.showTextArea(this.getReBackTranslationTextArea());
}
}
public void setReBackTranslationInModifying(int number, String mes){
if(number == this.core.getId()){
this.getReBackTranslationInModifyingTextArea().setText(mes);
this.showTextArea(this.getReBackTranslationInModifyingTextArea());
}
}
public void getTranslation(int number){
this.hideTextArea(this.getTranslationTextArea(),
this.core.getLanguageProperty("perspective.main.process.waiting.translation"));
this.core.getTranslation(number);
}
public void getOriginal(int number){
this.hideTextArea(this.getOriginalTextArea(),
this.core.getLanguageProperty("perspective.main.process.waiting.original"));
this.core.getOriginal(number);
}
public void getModifiedOriginal(int number){
this.hideTextArea(this.getModifiedOriginalTextArea(),
this.core.getLanguageProperty("perspective.main.process.waiting.modified_original"));
this.core.getModifiedOriginal(number);
}
public void getModifiedTranslation(int number){
this.hideTextArea(this.getModifiedTranslationTextArea(),
this.core.getLanguageProperty("perspective.main.process.waiting.modified_translation"));
this.core.getModifiedTranslation(number);
}
public void getModifiedTranslationForConfirmation(int number){
this.hideTextArea(this.getModifiedTranslationForConfirmationTextArea(),
this.core.getLanguageProperty("perspective.main.process.waiting.modified_translation"));
this.core.getModifiedTranslationForConfirmation(number);
}
public void getBackTranslation(int number){
this.hideTextArea(this.getBackTranslationTextArea(),
this.core.getLanguageProperty("perspective.main.process.waiting.back_translation"));
this.core.getBackTranslation(number);
}
public void getBackTranslationInModifying(int number){
this.hideTextArea(this.getBackTranslationInModifyingTextArea(),

```

```

this.core.getLanguageProperty("perspective.main.process.waiting.back_translation"));
this.core.getBackTranslationInModifying(number);
}
public void getReBackTranslation(int number){
this.hideTextArea(this.getReBackTranslationTextArea(),
this.core.getLanguageProperty("perspective.main.process.waiting.re_back_translation"));
this.core.getReBackTranslation(number);
}
public void getReBackTranslationInModifying(int number){
this.hideTextArea(this.getReBackTranslationInModifyingTextArea(),
this.core.getLanguageProperty("perspective.main.process.waiting.re_back_translation"));
this.core.getReBackTranslationInModifying(number);
}
public void getModifiedTranslationFirst(int number) {
this.hideTextArea(this.getModifiedTranslationFirstTextArea(),
this.core.getLanguageProperty("perspective.main.process.waiting.modified_translation"));
this.core.getModifiedTranslationFirst(number);
}
public void getModifiedTranslationNotFirst(int number) {
this.hideTextArea(this.getModifiedTranslationNotFirstTextArea(),
this.core.getLanguageProperty("perspective.main.process.waiting.modified_translation"));
this.core.getModifiedTranslationNotFirst(number);
}
public int getNumberOfTextArea(JTextArea area){
Component[] components = this.getComponents();
int count = 0;
for(Component c : components)
if(c.getClass().equals(JScrollPane.class)){
if(((JScrollPane)c).getViewport().getComponent(0).equals(area)) return count;
count++;
}
return -1;
}
public JTextArea getOriginalTextArea(){return this.areas[0];}
public JTextArea getModifiedOriginalTextArea(){return this.areas[2];}
public JTextArea getTranslationTextArea(){return this.areas[0];}
public JTextArea getModifiedTranslationForConfirmationTextArea(){return this.areas[0];}
public JTextArea getModifiedTranslationTextArea(){return this.areas[1];}
public JTextArea getBackTranslationTextArea(){return this.areas[1];}
public JTextArea getBackTranslationInModifyingTextArea(){return this.areas[1];}
public JTextArea getReBackTranslationTextArea(){return this.areas[1];}
public JTextArea getReBackTranslationInModifyingTextArea(){return this.areas[1];}
public JTextArea getModifiedTranslationFirstTextArea() {return this.areas[1];}
public JTextArea getModifiedTranslationNotFirstTextArea() {return this.areas[2];}
public ButtonGroup getButtonGroup(){return this.buttonGroup;}
public JRadioButton[] getRadioButtons() {return this.radios;}
public int getNumberOfValidTextAreas(){
for(int i=0;i<this.TEXT_AREA_NUMBER;i++) if(!this.textAreaPanels[i].isVisible()) return i;
return this.TEXT_AREA_NUMBER;
}
public int getSelectedRadioButtonNumber(){
for(int i=0;i<this.RADIOBUTTON_NUMBER;i++) if(this.radios[i].isSelected()) return i;
return -1;
}

```

```

}
public int getNumberOfValidRadioButtons(){
for(int i=0;i<this.RADIOBUTTON_NUMBER;i++) if(!this.radios[i].isVisible()) return i;
return this.RADIOBUTTON_NUMBER;
}

public JPanel createLoadingPanel(){
JPanel loadPanel = new JPanel();
loadPanel.setPreferredSize(new Dimension(this.TEXT_AREA_WIDTH, this.TEXT_AREA_HEIGHT));
GridBagLayout gridbag = new GridBagLayout();
GridBagConstraints c = new GridBagConstraints();
c.gridwidth = GridBagConstraints.REMAINDER;
loadPanel.setLayout(gridbag);
JLabel imageLabel = new JLabel(new ImageIcon("img/ajax-get-sentence.gif"));
gridbag.setConstraints(imageLabel, c);
loadPanel.add(imageLabel);
JLabel textLabel = new JLabel();
gridbag.setConstraints(textLabel, c);
loadPanel.add(textLabel);
return loadPanel;
}

public void showTextArea(JTextArea area){
int num = this.getNumberOfTextArea(area);
this.textAreaPanes[num].setVisible(true);
this.loadingPanel[num].setVisible(false);
this.validate();
}

public void hideTextArea(JTextArea area, String message){
int num = this.getNumberOfTextArea(area);
this.textAreaPanes[num].setVisible(false);
this.loadingPanel[num].setVisible(true);
((JLabel)this.loadingPanel[num].getComponent(1)).setText(message);
this.validate();
}

public void setValidTextAreas(int num, String[] labelTexts){
if(num != labelTexts.length) return;
if(num < this.getNumberOfValidTextAreas())
for(int i=this.getNumberOfValidTextAreas()-1;i>=num;i--){
this.textAreaPanes[i].setVisible(false);
this.labels[i].setVisible(false);
}
else if(num > this.getNumberOfValidTextAreas())
for(int i=this.getNumberOfValidTextAreas();i<num;i++){
this.textAreaPanes[i].setVisible(true);
this.labels[i].setVisible(true);
}
this.validate();
for(int i=0;i<num;i++) this.labels[i].setText(labelTexts[i].replaceAll(this.LINEFEED_REGEX, "\n"));
}

```

```

public void setValidTextAreas(int num){
    if(num < this.getNumberOfValidTextAreas())
        for(int i=this.getNumberOfValidTextAreas()-1;i>=num;i--){
            this.textAreaPanels[i].setVisible(false);
            this.labels[i].setVisible(false);
        }
    else if(num > this.getNumberOfValidTextAreas())
        for(int i=this.getNumberOfValidTextAreas();i<num;i++){
            this.textAreaPanels[i].setVisible(true);
            this.labels[i].setVisible(true);
        }
    this.validate();
}

public void setValidRadioButtons(int num, String[] commands){
    if(num != commands.length) return;
    if(num < this.getNumberOfValidRadioButtons())
        for(int i=this.getNumberOfValidRadioButtons()-1;i>=num;i--){
            this.radios[i].setVisible(false);
        }
    else if(num > this.getNumberOfValidRadioButtons())
        for(int i=this.getNumberOfValidRadioButtons();i<num;i++){
            this.radios[i].setVisible(true);
        }
    for(int i=0;i<num;i++) this.radios[i].setText(commands[i].replaceAll(this.LINEFEED_REGEX, "\n"));
    this.validate();
}

public void setValidRadioButtons(int num){
    if(num < this.getNumberOfValidRadioButtons())
        for(int i=this.getNumberOfValidRadioButtons()-1;i>=num;i--){
            this.radios[i].setVisible(false);
        }
    else if(num > this.getNumberOfValidRadioButtons())
        for(int i=this.getNumberOfValidRadioButtons();i<num;i++){
            this.radios[i].setVisible(true);
        }
    this.validate();
}

public void setButtonMouseAction(OnMouseAction adapter){
    MouseListener[] listeners = this.button.getMouseListeners();
    for(MouseListener l : listeners) this.button.removeMouseListener(l);
    this.button.addMouseListener(adapter);
}

public void setRadioButtonsMouseAction(OnMouseAction[] adapters){
    for(int i=0;i<adapters.length;i++){
        MouseListener[] listeners = this.radios[i].getMouseListeners();
        for(MouseListener l : listeners) this.radios[i].removeMouseListener(l);
        this.radios[i].addMouseListener(adapters[i]);
    }
}

public void setButtonVisible(boolean isVisible){
    if(this.button == null) return;
    this.button.setVisible(isVisible);
}

```

```

this.validate();
}

public void setButtonVisible(boolean isVisible, String label){
if(this.button == null) return;
this.button.setVisible(isVisible);
this.button.setText(label);
this.validate();
}

public int getId(){
return this.core.getId();
}

public Client getCore(){
return this.core;
}

public void setTextAreaEditable(int num, boolean isEditable){
this.areas[num].setEditable(isEditable);
if(isEditable) this.areas[num].setBackground(this.EDITABLE_COLOR);
else this.areas[num].setBackground(this.UNEDITABLE_COLOR);
this.validate();
}

public void setMessageVisible(boolean isVisible, String message, Color color){
this.generalMessage.setVisible(isVisible);
this.generalMessage.setText(message);
this.generalMessage.setForeground(color);
this.validate();
}

public void setMessageVisible(boolean isVisible, String message){
this.generalMessage.setVisible(isVisible);
this.generalMessage.setText(message);
this.generalMessage.setForeground(Color.BLACK);
this.validate();
}

public void setMessageVisible(boolean isVisible){
this.generalMessage.setVisible(isVisible);
this.validate();
}

public void setDefaultMessage(String message){
this.getCore().getMessageView().setDefaultText(message);
}

public void setView(int workCode, int id){
this.core.setId(id); // コアの現在編集集中の ID をセット
Protocol.setProcessView(workCode, id, this);
}

```

```

public void setWaitingView(){
Protocol.setWaitingView(this);
}

public boolean isWaitingView(){
if(this.getNumberOfValidTextAreas() == 0) return true;
else return false;
}

public void setFinishedView(){
Protocol.setFinishView(this);
}

public void notifyPartnerAction(Pair<Integer,Integer> lastAction){
if(lastAction != null)
this.setMessageVisible(true, this.core.getLanguageProperty("workcode.simple."+lastAction.getSecond(),
lastAction.getFirst()+1));
}
}

```

### **A.2.16 JMessageView.java**

```

package collaborative.client;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class JMessageView extends JPanel {
protected Client core = null;
protected JMainPerspective parent = null;
protected JTextArea messageArea = null;
protected JScrollPane messageScroll = null;
protected String defaultMessage = null;
private static final long serialVersionUID = 2542055428871315998L;
protected final int TEXT_AREA_WIDTH = 230;
protected final int TEXT_AREA_HEIGHT = 200;
protected final int LABEL_WIDTH = 230;
protected final int LABEL_HEIGHT = 20;
protected final String LINEFEED_REGEX = "\\[\linefeed\\]";

public JMessageView(JMainPerspective parent, Client core){
super();
this.parent = parent;
this.core = core;
this.make();
}

public void make(){
GridBagLayout gridbag = new GridBagLayout();

```

```

GridBagConstraints c = null;
this.setLayout(gridbag);
c = new GridBagConstraints();
c.gridwidth = GridBagConstraints.REMAINDER;
JLabel label = new JLabel(this.core.getLanguageProperty("perspective.main.message.title"));
label.setPreferredSize(new Dimension(this.LABEL_WIDTH,this.LABEL_HEIGHT));
gridbag.setConstraints(label, c);
this.add(label);
c = new GridBagConstraints();
this.messageArea = new JTextArea();
this.messageArea.setLineWrap(true);
this.messageArea.setWrapStyleWord(true);
this.messageArea.setEditable(false);
this.messageArea.setBackground(Color.WHITE);
this.messageScroll = new JScrollPane(this.messageArea);
this.messageScroll.setPreferredSize(new Dimension(this.TEXT_AREA_WIDTH, this.TEXT_AREA_HEIGHT));
gridbag.setConstraints(this.messageScroll, c);
this.add(this.messageScroll);
}

public void showText(String text){
if(text == null) text = "";
else text = text.replaceAll(this.LINEFEED_REGEX, "\n");
this.messageArea.setText(text);
}

public void showDefaultText(){this.messageArea.setText(this.defaultMessage);}

public void setDefaultText(String text){
if(text == null) this.defaultMessage = "";
else this.defaultMessage = text.replaceAll(this.LINEFEED_REGEX, "\n");
}
}

```

### **A.2.17 JStateView.java**

```

package collaborative.client;
import java.awt.Dimension;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class JStateView extends JPanel {
protected Client core = null;
protected JMainPerspective parent = null;
protected JLabel sentenceNumberLabel = null;
protected JLabel modifyNumberLabel = null;
protected int numberOfSentences = 0;
private static final long serialVersionUID = 4392497596823849528L;
protected final int LABEL_WIDTH = 230;
protected final int LABEL_HEIGHT = 20;

public JStateView(JMainPerspective parent, Client core){

```



```

super();
this.parent = parent;
this.core = core;
this.make();
}

public void make(){
GridBagLayout gridbag = new GridBagLayout();
GridBagConstraints c = new GridBagConstraints();
c.gridwidth = GridBagConstraints.REMAINDER;
this.setLayout(gridbag);
this.sentenceNumberLabel = new JLabel();
this.sentenceNumberLabel.setPreferredSize(new Dimension(this.LABEL_WIDTH,this.LABEL_HEIGHT));
gridbag.setConstraints(this.sentenceNumberLabel, c);
this.add(this.sentenceNumberLabel);
this.modifyNumberLabel = new JLabel();
this.modifyNumberLabel.setPreferredSize(new Dimension(this.LABEL_WIDTH,this.LABEL_HEIGHT));
gridbag.setConstraints(this.modifyNumberLabel, c);
this.add(this.modifyNumberLabel);
}

public void setSentenceNumber(int id){
String text = id + this.core.getLanguageProperty("perspective.main.state.sentence", this.numberOfSentences);
this.sentenceNumberLabel.setText(text);
}

public void setModifyCount(int count){
String text = this.core.getLanguageProperty("perspective.main.state.modify_count")+": "+
count+this.core.getLanguageProperty("perspective.main.state.count");
this.modifyNumberLabel.setText(text);
}

public void setNumberOfAllSentences(int number){this.numberOfSentences = number;}
public void setNoSentencesWorked(){
String textSentence = "-" +
this.core.getLanguageProperty("perspective.main.state.sentence", this.numberOfSentences);
this.sentenceNumberLabel.setText(textSentence);
String textModify = this.core.getLanguageProperty("perspective.main.state.modify_count")+": "
+"-"+this.core.getLanguageProperty("perspective.main.state.count");
this.modifyNumberLabel.setText(textModify);
}
}

```

### **A.2.18 JSubMenu.java**

```

package collaborative.client;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;

public class JSubMenu extends JMenuBar implements ActionListener {
protected Client core = null;
protected JClientFrame parent = null;

```

```

protected JMenuItem saveItem = null;
protected JMenuItem exitItem = null;
protected JMenuItem configItem = null;
private static final long serialVersionUID = 629690974476041794L;

public JSubMenu(Client core, JClientFrame parent){
    super();
    this.core = core;
    this.parent = parent;
    this.make();
}

public void make(){
    this.add(this.getFileMenu());
    this.add(this.getConfigMenu());
}

public JMenu getFileMenu(){
    JMenu fileMenu = new JMenu(this.core.getLanguageProperty("frame.submenu.file"));
    this.saveItem = new JMenuItem(this.core.getLanguageProperty("frame.submenu.file.save"));
    this.exitItem = new JMenuItem(this.core.getLanguageProperty("frame.submenu.file.exit"));
    this.exitItem.addActionListener(this);
    fileMenu.add(this.saveItem);
    fileMenu.addSeparator();
    fileMenu.add(this.exitItem);
    return fileMenu;
}

public JMenu getConfigMenu(){
    JMenu configMenu = new JMenu(this.core.getLanguageProperty("frame.submenu.config"));
    this.configItem = new JMenuItem(this.core.getLanguageProperty("frame.submenu.config.config"));
    configMenu.add(this.configItem);
    return configMenu;
}

public void actionPerformed(ActionEvent ae) {
    if(ae.getSource().equals(this.exitItem)){
        String obj = core.getLanguageProperty("perspective.main.frame.exit_confirm.1");
        if(JOptionPane.showConfirmDialog(parent, obj,
            core.getLanguageProperty("perspective.main.frame.exit_confirm.title"),
            JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE) == JOptionPane.YES_OPTION)System.exit(0);
    }
}
}

```

### **A.2.19 Sentence.java**

```

package collaborative.client;

public class Sentence {
    protected int workCode = 0;
    protected int modifyCount = 0;

    public Sentence(int workCode){this.workCode = workCode;}
}

```

```

public int getWorkCode(){return this.workCode;}
public int getModifyCount(){return this.modifyCount;}
public void setWorkCode(int workCode){this.workCode = workCode;}
public void setModifyCount(int modifyCount){this.modifyCount = modifyCount;}
public void incrementCount(){this.modifyCount++;}
public boolean isUnderLimitedCount(int limitedCount){return this.modifyCount < limitedCount;}
public void resetCount(){this.setModifyCount(0);}
}

```

### A.2.20 ComboBoxPair.java

```

package collaborative.client;
import collaborative.common.Pair;

public class ComboBoxPair<T,U> extends Pair {
public ComboBoxPair(T first, U second){
this.first = first;
this.second = second;
}

public void setKey(T obj){this.setFirst(obj);}
public void setValue(U obj){this.setSecond(obj);}
public Object key(){return this.getFirst();}
public Object value(){return this.getSecond();}
public String toString(){return (String)this.value();}
}

```

### A.2.21 OnMouseAction.java

```

package collaborative.client;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import javax.swing.JRadioButton;

public class OnMouseAction implements MouseListener {
public JProcessView processView = null;
public JMessageView messageView = null;
public String message = null;

public OnMouseAction(){
public OnMouseAction(JProcessView pane){
this.messageView = pane.getCore().getMessageView();
this.processView = pane;
}

public OnMouseAction(JProcessView pane, String mes){
this(pane);
this.message = mes;
}

public void mouseEntered(MouseEvent arg0){this.messageView.showText(this.message);}
public void mouseClicked(MouseEvent arg0) {
if(arg0.getSource().getClass().equals(JRadioButton.class))((JRadioButton)arg0.getSource()).setSelected(true);
}

public void mousePressed(MouseEvent arg0) {}
public void mouseReleased(MouseEvent arg0) {}
public void mouseExited(MouseEvent arg0) {}
}

```

```
}
```

### A.2.22 Pair.java

```
package collaborative.common;
public class Pair<T, U>{
    public Pair(){
    public Pair(T first, U second){this.first = first;this.second = second;}
    public T getFirst(){return first;}
    public U getSecond(){return second;}
    public void setFirst(T first){this.first = first;}
    public void setSecond(U second){this.second = second;}
    public boolean hasFirst(){
        if(this.first == null) return false;
        else return true;
    }
    public boolean hasSecond(){
        if(this.second == null) return false;
        else return true;
    }
    public boolean equals(Object value){
        if(value == null) return false;
        if(!getClass().equals(value.getClass())) return false;
        return equals((Pair<?, ?>)value);
    }
    public boolean equals(Pair<?, ?> value){
        if(value == null) return false;
        if(!first.equals(value.getFirst())) return false;
        if(!second.equals(value.getSecond())) return false;
        return true;
    }
    public int hashCode() {int h = 0;h = first.hashCode();h = h * 31 + second.hashCode();return h;}
    public String toString() {
        StringBuilder b = new StringBuilder();b.append("[");b.append(getClass().getName());
        b.append(" ");b.append("first:");b.append(first.toString());b.append("second:");
        b.append(second.toString());b.append("]");return b.toString();
    }
    public static <T, U> Pair<T, U> create(T v1, U v2){return new Pair<T,U>(v1, v2);}
    protected T first;
    protected U second;
}
```