

分散協調翻訳支援システムの構築と運用

小山 博史^{†1} 風間 一 洋^{†2}
岡本 隆史^{†3} 横田 健彦^{†4}

Ja-Jakarta プロジェクトはオープンソースプロダクトの日本語化やドキュメントの翻訳をする活動をしており、オープンソースモデルに基づいた翻訳体制で分散協調翻訳をしている。ただし、分散協調翻訳は必ずしも効率的ではなく、高い品質を確保できるとは限らない。本論文では、創発的な効果ができるだけ得られるような分散協調翻訳体制を示す。次に、その下で構築・運用した分散協調翻訳支援システムと、翻訳作業におけるメタデータの利用について述べる。さらに、運用したシステムのログや翻訳者へのアンケートなどを分析し、実際に効果があることを示す。

Construction and Operation of Distributed Cooperative Translation Support System

HIROSHI KOYAMA,^{†1} KAZUHIRO KAZAMA,^{†2} TAKASHI OKAMOTO^{†3}
and TAKEHIKO YOKOTA^{†4}

The objectives of the Ja-Jakarta project are to localize open source products and to translate their documentations into Japanese. The organization based on the open source model is aimed to establish a framework of distributed cooperative translation. However distributed cooperative translation is not always effective and it is difficult to keep the quality of translation. In this paper, we explain the organization which produces effective cooperation, the construction and operation of our translation support system, and the usage of metadata for translation. And we show the effectiveness in improving the productivity of our organization by analyzing the system logs and the questionnaires to translators.

1. はじめに

近年、オープンソースソフトウェアは商用ソフトウェアと機能や品質が対等なレベルまで向上し、企業の業務システム構築に不可欠になりつつある¹⁾。しかし、大部分のオープンソースプロジェクトは欧米から生まれることが多く、それを母語が異なる日本で使用するためには、ソフトウェアの国際化作業や日本語化作業、およびドキュメントの翻訳を行わなければならない。このために、オープンソースプロジェクトの内外に、

このような地域固有の作業に関する活動が存在する。その1つに、Javaに関するオープンソースソリューションを提供するJakartaプロジェクトの日本固有の作業を行うJa-Jakartaプロジェクトがある。これは2000年9月に発足し、日本語化・国際化などに関してはJakartaプロジェクトと共同で作業しているが、ドキュメント翻訳に関しては共同作業が困難なために、独自にオープンソースモデルに基づいた翻訳体制で分散協調翻訳をしている。

ただし一般的には、分散協調体制によって作業の効率化もしくは作業品質の向上を望めるとは必ずしもいえない。たとえば、Lorgeらは単純な論理課題を用いた実験の結果を評価し、協調課題遂行がグループの平均的なメンバの課題遂行能力を上回るにしても、グループの最良メンバのパフォーマンスには及ばない場合が多いことを示した²⁾。またLaughlinらは、トランプを用いた帰納課題を用いて同様の結果を得た³⁾。

その一方で、Okadaらは科学的仮説形成のプロセスに関する研究において、協調活動では競合仮説を思い浮かべたり仮説の根拠について考察したりする「説明

^{†1} 長野県工科大学情報技術科
Computer and Systems Engineering, Nagano Prefectural Institute of Technology

^{†2} 日本電信電話株式会社未来ネット研究所
Network Innovation Laboratories, Nippon Telegraph and Telephone Corporation

^{†3} 株式会社 NTT データ技術開発本部
Research and Development Headquarters, NTT Data Corporation

^{†4} 株式会社東芝研究開発センター
Corporate Research & Development Center, Toshiba Corporation

活動」に関わる頻度が高く、これが問題解決のための有効な要因になっていることを明らかにするなど、協調活動がグループの創発性を誘発する効果があることを実験で確かめた⁴⁾。なお創発性とは、初期状態と発展規則だけでは、システムがどのような状態へと発展するのは非自明であり、かつ実際に時間発展させた結果として得られる状態が意味づけ可能であることである。また植田らは、グループのメンバ個々人が持つ知的資源の単なる総和以上の効果がグループのレベルで創発する可能性を、企業の研究所における研究・開発の現場での協調活動を対象として調査分析し、実際に新たな知識やアイデアが創発したことを示した⁵⁾。さらに、課題が概念変化 (conceptual change)⁶⁾ をともなうような創造的なものである場合、つまり説明活動がその課題の達成に大きく寄与する場合は、説明活動を頻繁に行えるような条件がグループに備わっていれば、グループが創発性を発揮する可能性が十分であると分析した。

翻訳作業は新たな知識やアイデアを生み出すことが必要な課題ではないが、自然で分かりやすい翻訳文を作成するためには、いったん英語で書かれた文章を完全に理解しなければならない、原文の間違いを発見して修正しなければならない、さらにまったく異なる文化を背景に持つ日本語の文章を新たに作成しなければならないなどの点で創造的な側面を持つ。そのため分散協調翻訳においても、説明活動を頻繁に行えるような体制を構築することでポジティブな効果が創発することは十分期待できる。そこで、我々はポジティブな効果が創発しやすくなるようなプロジェクト運営とシステム構築を行った。具体的には、翻訳作業においてメンバが説明活動を頻繁に行えるように、メンバ同士ができる限り対等であるような体制づくりを行った。また、説明活動が円滑に行われるように、説明活動の場であるメーリングリストを「翻訳作業用」、「プロジェクト運営用」、「一般議論用」のように用途に応じて複数用意した。この結果、システム構築時には存在しなかったメタデータ付与による翻訳スキームが提案され実現されるなど、ポジティブな効果が実際に創発したことを確認した。

本論文の構成は以下のとおりである。2章では関連研究について、3章では分散協調翻訳の必要性と問題点について述べる。4章では、オープンソースモデルに基づいた翻訳体制を紹介し、5章では実際に構築した分散協調翻訳支援システムを紹介し、6章では翻訳におけるメタデータ付与方法と、その活用法について述べる。7章では、運用ログや翻訳者のアンケートを

分析した結果について述べる。最後に、8章で今後の課題を、9章でまとめを述べる。

2. 関連研究

2.1 オープンソースモデル

Raymond は Linux などのオープンソースプロジェクトで採用されているソフトウェア開発形態をバザール方式と呼び、優秀な開発者とテストを得ることができる⁷⁾。一般に、これをオープンソースモデルと呼ぶ。ただし、このモデルは必ずしもソフトウェア開発だけに用いられるとは限らず、Berglund らはドキュメント作成に適用することを提唱している⁸⁾。本論文では、翻訳への適用事例について述べる。

2.2 分散協調翻訳

高橋は、マルチメディア教材をネットワーク上で共同で翻訳している⁹⁾。この報告では、翻訳管理のために事務局を設置し、一元的な管理を行っている。しかし、企業のソフトウェア開発者が中核となる場合には、このような管理負荷が高い体制を維持することは難しい。そこで本論文では、管理主体を分散化させるとともに、翻訳作業を自律的に行うことで管理負荷を減少させる手法について述べる。

2.3 メタデータの利用

メタデータの利用例として、注釈の付加がある。たとえば、Annotea プロジェクト¹⁰⁾では、XPointer¹¹⁾を用いて、原文を修正せずに注釈を RDF¹²⁾ 形式で追加している。本論文の手法では、原文に翻訳作業で有用なメタデータを付加する。Annotea ではメタデータを参照して原文とオントロジーデータベースの注釈がハイパーリンクされたビューを生成するのに対し、本論文の手法ではメタデータを参照して、目的用途に応じた複数のビューを生成し分ける点が異なる。

また、今村らはメタデータの付与コストの低減が Semantic Web の普及の鍵であると論じている¹³⁾。その中で、業務で役立つメタデータを作成するコストは一般に高いことから、メタデータ作成用のエディタや半自動生成ツールが多数開発されつつある状況を説明している。このように、人間によるメタデータの付与コストは課題としてあげられることが多い。本論文では、メタデータの付与により作業員自身も利益を得ることができる支援システムを構築すれば、人手で付与する場合でも積極的に受け入れられる事例を示す。

3. 分散協調翻訳

3.1 翻訳作業の課題

プロジェクト発足当初は互いの成果を統合して公開

するだけであったが、次第に翻訳作業が次のような特徴を持つことが明らかになった。

- 日本では、企業のオープンソース活動に関する認知度が低いために、個人的な余暇を活用した副次的な活動にならざるをえない。
- Jakarta プロジェクトは、一般ユーザではなく開発者が使用するプロダクトがほとんどであることから、参加者は開発者が大部分であり、学生や研究者は少ない。そのため、本質的に作業時間にとりにくく、しかも作業量とスケジュールが本業の忙しさに大きく左右される。
- オープンソース活動に対して金銭的な寄付が得にくいので、職業翻訳者を雇用するのは難しく、ボランティアだけで対応せざるをえない。

これから、オープンソースの翻訳活動では翻訳者を安定して確保することが困難であることが分かる。しかし、昨今は企業システムにオープンソースプロダクトが組み込まれるようになったことから、継続的な翻訳更新の要求も高く、翻訳者の確保は、より深刻な課題となっている。

3.2 分散協調翻訳の導入

特定の翻訳者に長期的・継続的に安定して参加してもらうことが困難であるのならば、短期的な活動であってもかまわないので、できる限り多くの翻訳者に参加してもらうことを考え、分散協調翻訳を導入した。

本論文では、分散協調翻訳を「複数の翻訳者がネットワークで互いに情報を交換したり仕事の割当てを行いながら協力しあって単一ドキュメントの翻訳作業を行うこと」と定義する。ただし、対象とする電子化ドキュメントでは HTML 形式が多用されるが、複数の HTML ファイルがハイパーリンクで結合されるために、必ずしもドキュメントというまとまりが明確ではない。そこで、「Tomcat 5.0 ドキュメント」のような意味的まとまりを持ち、同一アーカイブに収録されるファイル群を単一のドキュメントと見なす。

ここで重要なのは、互いが自発的に協力し合って、翻訳作業だけでなく管理作業の負荷も分散させることである。すでに述べたように、本プロジェクトでは各参加者はボランティアとして余暇を活用して作業しており、各自が割ける時間は少ないうえで一定しない。したがって、一般の書籍翻訳のように翻訳作業の分担、訳語の選択、進捗状況を管理するマネージャを置いてトップダウンに作業する体制では、管理作業の負荷が特定の人物に一極集中してしまい、個人さらには全体の作業停滞を引き起こしやすい。

そこで、翻訳者が互いに情報交換をしたり、自分が

ら作業割当てを申し出たりすることでネットワーク上で分散している翻訳者たちが自律的に作業を進めていける体制を構築している。本論文では、このような管理主体の分散化と翻訳作業の自律化を兼ね備えた翻訳作業について論じる。

なお、管理作業を分散化すると、特定個人が強いリーダーシップを発揮することは難しくなり、翻訳作業全体を適切に進行できなくなる危険性が存在する。しかし、作業に関する議論を活性化して、説明活動を頻繁に行うことができるならば、最初に意図していなかったポジティブな効果が創発的に生まれることが期待でき、作業効率改善につながる体制やシステムの改善が行われると推測される。

3.3 分散協調翻訳の問題とその解決

ただし、活動が進行するにつれて、分散協調翻訳には、作業効率の低下や翻訳品質の低下へ直結する次のような問題が存在することが明らかになった。

- 翻訳担当が決まったファイルの翻訳作業が停滞すると、全体の作業が完了できない。
- 翻訳者が自分から作業の停滞と担当替えを言い出すことが少なく、自律的な再割当てが難しい。
- 翻訳者に翻訳作業を理解してもらおう時間が、参加の障壁になりやすい。
- 翻訳グループごとに、局所的な規則ができやすく、全体の統一が難しい。
- 分散作業のために意思疎通が難しく、文体、表現、表記、訳語が統一されにくい。
- 頻繁な原文更新についていくためには、大胆な意訳がしにくいので、直訳に近くなってしまう。

ボランティアによる分散協調翻訳では、翻訳者は必ずしもすでに翻訳経験があるわけではなく、1人の翻訳者が翻訳できるファイルの量も限られる。そのため、少数の経験豊かな翻訳者を専任で割り当てて一気に処理できる商業翻訳のような短時間で高品質の翻訳成果を得ることは期待できないが、オープンソースプロダクトのドキュメントの日本語訳の要求は高く、必ずしも高品質でなくても、ある程度妥当な品質であれば有益であると判断される。

また、商業翻訳とは異なり、ドキュメントの翻訳が終了した部分から即座に公開したり、誤りを指摘されたときに即時に反映することも可能である。

そこで、前述の問題を完全に解決することを考えるのではなく、作業効率を改善し、ある程度妥当な品質の翻訳ドキュメントをできるだけ早く公開するという視点から次の2種類の対策を行った。

- 翻訳者の自由度を確保するとともに、協調行動に

よる創発的な効果が得られる組織にすることで、翻訳作業の負荷分散を図った。

- CVS¹⁴⁾、Web や電子メールを活用したサーバ中心型の分散協調翻訳支援システムを構築することで、翻訳者間の情報共有の促進と協調作業負荷の軽減とを図った。

さらに、実際の活動の中で、翻訳作業の負荷を軽減するために、次のような翻訳スキームが実現された。

- 翻訳時にメタデータを付加し、それを変換することで、目的に応じたドキュメントを生成する。

これは、特定の人物が提案したのではなく、翻訳作業を続けながら、複数人間によって創発的に提案、実装されたものである。

この後の節で、これらについて、さらに詳しく説明する。

4. 翻訳体制

翻訳体制で重要なことは、自発的に参加してもらうボランティアの士気を高めることで参加人数と活動の継続性を確保し、さらに協調作業の創発的な効果を得ることと、組織としての意志決定やセキュリティ管理を確実にを行うために役割分担を明確にして、身分が保証されていない参加者にはそれらの権利を与えないことという、互いに相反する要素を両立させることである。

そこで、我々は翻訳作業においてはできる限り対等な人間関係を確認し、翻訳管理作業においては各人の役割と責任が明確な組織になるよう配慮した。

4.1 組織の構成

翻訳管理作業において、参加者の役割と責任分担を明確にするために、開発者、コミッタ、リリースマネージャ、システム管理者、プロジェクト運営委員会委員 (PMC)、PMC 議長の役割を設置した。なお、ここで開発者と呼ぶのはプログラムの国際化などのソフトウェア開発も行っているからであるが、本論文では翻訳作業に限定するので、以降は翻訳者と記す。

翻訳作業は、Jakarta プロジェクトのサブプロジェクトごとに翻訳サブプロジェクトを作成し、それに参加する翻訳者はメーリングリスト (ML) で情報交換しながら翻訳する。ただし、導入された翻訳規則や開発された翻訳用ツールなどは、複数の翻訳サブプロジェクト間で共有される。

翻訳進捗状況表や翻訳ファイルは、十分な貢献を持つ翻訳者の中からサブプロジェクトごとに投票で選出された複数のコミッタが分散管理し、オフィシャルサーバで公開する。コミッタは、プロジェクトの将来

に関係する事項に関する投票権も持つ。

翻訳進行や成果物の公開については、コミッタより選出された1名のリリースマネージャが一括管理する。

翻訳活動支援システムの管理は、コミッタより選出された複数のシステム管理者により実施される。

プロジェクトの戦略的指針と成功に関して責任を負うのは、コミッタより選出された PMC である。PMC から選出された PMC 議長は、PMC の議事進行や対外折衝に責任を負う。

このように、貢献度の高い人材をより責任ある立場へ推薦し、強い権限を与える実績主義を採用した。基本的に、Apache Software Foundation (ASF) の体制¹⁵⁾ の影響を強く受けているが、翻訳管理作業においては役割と責任分担を明確にする半面、翻訳作業に関してはコミッタも翻訳者もほぼ同等の立場である点が特長的である。たとえば、ASF では、プログラムの主要な機能のほとんどをコミッタが実装し、開発者はバグの報告や修正パッチの送付を行う程度にすぎない。

4.2 翻訳の手順

では、具体的な翻訳の手順について説明する。

翻訳作業はファイル単位で行い、着手可否評価中、翻訳者募集中、翻訳中、初稿 (校正者募集中)、校正中、翻訳者校正中、校正者校正判定中、校了という8種類の翻訳進捗状態をファイルごとに管理する。翻訳作業が進行するにつれ、翻訳進捗状態はこの順に遷移する。

実際の翻訳作業は、翻訳対象ファイルの割当てから開始する。まず翻訳者は、翻訳者募集中の状態を持つファイル群から自分が翻訳したいファイルを選択する。翻訳者がコミッタである場合には、翻訳進捗管理表を更新する。翻訳者がコミッタでない場合には、翻訳者は ML にそのファイルを担当したい意志を宣言し、コミッタはそれを受けて翻訳進捗管理表を更新する。ここで、翻訳進捗管理表の更新とは、担当者登録と、対象ファイルの状態を更新することである。これで、翻訳担当者の割当てが完了する。割当てを自律分散して行えるようにしたことで、時間を多く確保できる翻訳者にはより多くのファイルを担当してもらい、まだ慣れていないまたは時間をあまり確保できない翻訳者には1ファイルからでも担当してもらうことができる。

その後、翻訳担当者は翻訳対象ファイルをサーバから取得して、翻訳を行う。翻訳が終了したら、翻訳担当者がコミッタである場合には直接サーバの CVS に初稿ファイルを登録し、コミッタでない場合にはコミッタに登録を依頼する。コミッタは依頼された初稿ファイルを CVS に登録する。コミッタは同時に翻訳進捗

管理表の更新も行う。

さらに、初稿ファイルに対して、翻訳担当者の割当てと同じ方法で校正担当者を割り当て、校正担当者はそのファイルを CVS から取得して校正する。ただし、翻訳品質を向上させるために、誤訳や変換ミスの訂正は、校正担当者に限らず、そのファイルを閲覧または利用する他の翻訳者によっても行われる。校正が終了すると、確認のために翻訳者校正、校正者校正判定という手順を踏んで校了となる。これらの手順を踏む場合にも、対象ファイルの取得、CVS への登録は翻訳時と同様である。翻訳進捗管理表については、対象ファイルの状態のみ更新する。

そして、すべてのファイルの状態が校了になったときに、翻訳の終了である。その後リリースマネージャが配布用パッケージを作成して、Web で公開する。

なお、翻訳作業全体を通して、原文のミスの指摘、翻訳文に関するコメントの募集、訳語の選択などの議論は随時行われ、その結果は翻訳者またはコミッタによって反映される。

4.3 翻訳の管理

実際の翻訳作業とは別に、若干の翻訳管理作業が必要になるが、これは複数のコミッタまたはリリースマネージャなどの、すでに参加実績があり、オフィシャルサーバに直接アクセスできる権限を持つ参加者によって行われる。たとえば、管理作業としては翻訳進捗およびその管理表の管理と、翻訳ファイルの CVS への登録、翻訳者へのファイル登録作業終了通知、配布用パッケージの作成などがあげられる。

実際には、翻訳管理作業は翻訳作業よりも少ない人数で行わなければならないので、ここがボトルネックになりやすく、一部の参加者が本業が多忙になったなどの理由で活動できる時間が少なくなると、翻訳作業全体が停滞してしまう現象も多く観察されている。つまり、翻訳管理作業の負荷をどの程度削減できるかは、翻訳作業の成否を大きく左右すると考えられる。

5. 分散協調翻訳支援システム

5.1 分散協調翻訳支援

活動初期は、ML の運用や成果を Web で公開するために企業のサーバを借用していたが、分散協調翻訳に関しては、参加者間のファイルや情報共有の困難さ、作業用ツールの不統一などの問題が健在化し、以下のような要求が出てきた。

- データの修正や引き継ぎを容易にするために、いままでも個人または複数のサーバで分散管理していた作業データ、ML アーカイブおよび Web デー

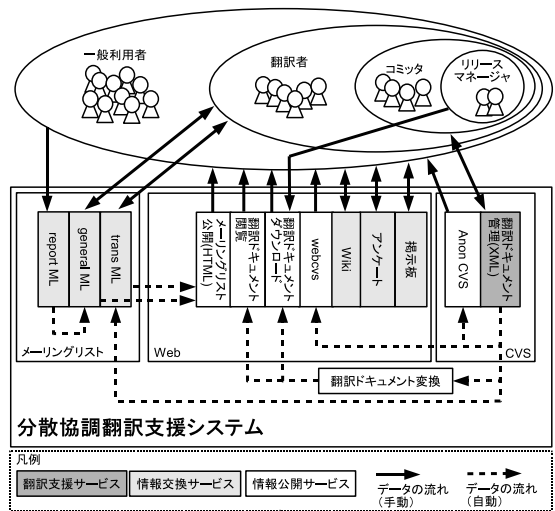


図 1 サービス構成図
Fig.1 Service structure.

データを、1 カ所に集中させて、参加者間で共有できるようにする。

- 翻訳作業に必要なプログラムをある程度共通化することで、翻訳作業環境の統一を図り、ノウハウの共有を容易にする。
- 翻訳作業用ツール、ML プログラムおよび Web サーバを相互連携させることで作業の自動化を図り、翻訳者の負荷を軽減する。
- 翻訳作業と並行して、作業効率改善のために独自の翻訳作業用ツールを開発する。

そこで、オフィシャルサーバを設置し、そこにデータやリソースを集中するとともに、分散協調翻訳を支援するシステムを構築し、各種のサービスを提供することを試みた。なお当時でも SourceForge.jp などのオープンソース活動を総合的に支援する無償の支援サービスは提供されていたが、翻訳作業という特殊な分野向けに独自に開発したツールをシステムに統合する自由度までは与えられなかったために利用しなかった。

5.2 システムのサービス構成

本システムで提供するサービスは、大きく翻訳支援、情報交換、情報公開の 3 種類に分類できる。分散協調翻訳支援システムのサービス構成図を図 1 に示す。

翻訳支援サービスは、CVS による編集された翻訳ファイルの競合回避とバージョン管理、翻訳元データからのドキュメント生成などが含まれる。特に、CVS は競合回避のために利用者に排他的なアクセスを強要することはなく、コピー・修正・マージモデルに基づき競合解消のための手続きを真に必要なときまで遅延させるので、ファイルの同時編集が可能になり、

分散協調作業に対する親和性が高い。

情報交換サービスは、訳語統一や誤り修正などのための翻訳に関する意見交換と、分担決定や進捗報告などの翻訳管理情報の交換を目的とし、翻訳者がつねに情報が得られるように翻訳作業用メーリングリスト (trans ML) を設置している。他に、目的に応じて一般的議論のための general ML、フィードバックを投稿する report ML などの ML だけでなく、一般利用者が Jakarta プロダクトの利用法を議論する掲示板、イベント用の Wiki など提供している。

情報公開サービスは、翻訳ドキュメントや活動情報などを公開するサービスであり、Web サーバで行っている。他に、Anon CVS や cvsweb を用いて翻訳ファイルを公開したり、利用者の動向調査のために専用システムを作成してアンケート実施と連動した集計結果の公開を行っている。

5.3 サービスの連携と自動化

本論文で特徴的なことは、翻訳ファイルの更新をきっかけにして情報交換サービスや情報公開サービスに連携させるファイル指向の自動サービス連携を実現していることである。ファイルに対する翻訳や校正が終了し、CVS にコミットされることをきっかけに、以下が自動的に処理される。

- (1) trans ML に対する作業概要と差分の自動投稿。この自動投稿されるメールをコミットメールと呼ぶ。
- (2) 翻訳元データを変換し翻訳ドキュメントを自動生成。
- (3) 翻訳ドキュメントの自動公開。
- (4) 翻訳ドキュメントの自動アーカイブ作成。

この自動化は、一般的なシェルや perl に加えて、Ant¹⁶⁾ を用いてスクリプトを記述して行う。Ant は、Java で記述された構築ツールであり、XML や CVS の使用が最初から考慮されていること、Java で記述されたツールと親和性が高く、高速に起動できることなどの利点を持つ。

このような自動サービス連携により、翻訳者の負担が軽減されるだけでなく、プロジェクトに参加していない利用者にとっても成果が即座に反映されるために、つねに最新の成果を参照できる利点がある。

6. 翻訳におけるメタデータの利用

6.1 翻訳作業とメタデータ

プロジェクト活動当初は、HTML ファイルとテキストファイルが翻訳対象であった。しかし、XML¹⁷⁾ の普及とともに、XML ファイル形式で記述し、XSLT

表 1 翻訳作業で使用する独自タグ
Table 1 Original tags used for translation.

| 意味 | マニュアル用 | API リファレンス用 |
|--------|--------------|-------------|
| 翻訳者 | <translator> | @translator |
| 校正者 | <editor> | @editor |
| 翻訳状態 | <status> | @status |
| 翻訳更新日 | <update> | @update |
| 訳注 | <annotation> | @annotation |
| 訳者コメント | <note> | @note |
| 原文 | <primary> | @primary |

プロセッサ¹⁸⁾ で指定されたスタイルシートに基づいて HTML ファイルに変換する方法が主流になった。

また、プロジェクトの進行につれ、API ドキュメントの翻訳も手がけるようになった。Java では、Knuth の文芸的プログラミング (Literate programming)¹⁹⁾ の影響を受けて、ソースコード内にドキュメンテーションコメントを記述し、javac コンパイラでバイトコードを、javadoc コマンド²⁰⁾ で HTML 形式の API リファレンスを生成する。

なお、一般に訳文は単に原文を変換したものではなく、訳者に関する情報や、訳注が追加されていることが多く、原文と完全に 1 対 1 対応するわけではない。また、校正時には、原文・訳文の併記形式やコメントを記述できるようにダブルスペース形式など、異なる書式で印刷することも多い。

そこで、XML や Java ソースが構造化ドキュメントであり独自タグが使用できること、さらに元データを変換することにより対象ドキュメントを生成することに着目し、翻訳対象のファイルに独自タグを用いてメタデータや原文などの情報を付与し、さらにこのデータを利用することで翻訳進捗状態に応じて異なるビューを持つドキュメントを生成する。

6.2 独自タグの導入

現在、マニュアルおよび API ドキュメント記述用に使用している独自タグを表 1 に示す。

マニュアル用について説明すれば、<translator> タグ、<editor> タグおよび <status> タグは書誌情報に相当するメタ情報の記述に、<annotation> タグと <note> タグは注釈に相当するメタ情報の記述に用いる。注釈用が 2 種類存在するのは、<note> タグが翻訳者と校正者の間の情報共有にのみ用いるのに対し、<annotation> タグは訳者から読者への情報伝達に用いるからである。ただし、<primary> タグは例外であり、メタ情報ではなく、原文データと訳文データを同一ファイルに保持するために使用し、原文をこのタグで囲う。

6.3 ドキュメント変換

図 2 に、マニュアルの変換例を示す。同一の XML ファイルを変換して、原文・訳文併記ビューと訳文ビューの 2 種類のドキュメントを生成する。訳文と原文を併記すれば校正をしやすくなるが、一般読者は訳文しか読まない傾向があるので、公開・配布用には訳文のみを生成することで可読性を向上させる。

翻訳者、校正者は、クレジット情報としてドキュメ

ントに埋め込まれる。オープンソースプロジェクトで成果に貢献者の名前を明示することは、参加の強い動機付けになるために重要であり、活動を活性化するために役立つ。

訳注、訳者コメントもドキュメントの文中に埋め込まれる。ただし、訳者コメントは訳者と校正者だけが見ればよいので、訳文ビューでは表示されないことに注意されたい。

従来は、クレジットや訳文の挿入、原文の削除などは翻訳者・校正者が人手で行っていたが、この自動化により、作業負荷の低減とより作業に適した特殊なビューの提供が実現できた。

7. 評価

7.1 活動状況の分析

まず、最初に活動状況について分析する。

trans ML の参加人数は、2004 年 7 月 7 日の時点で 718 名である。コミッタ数は 22 名 (3.1%) で、このうち PMC 数は 8 名 (1.1%) である。さらに、2003 年 10 月から 2004 年 7 月までに翻訳に協力した人数 (コミッタに昇格した者を除く) は 17 名 (2.4%) である。trans ML におけるメール投稿数はコミットメールを含めて週平均 40.2 通であり、活発に活動していると考えられる。ただし、実際に作業に参加する人間の割合がかなり低い点が問題である。

さらに、翻訳作業に参加した人達に職業を尋ねたところ 19 名の回答を得て、その内訳はソフトウェア技術者 (84.1%)、コンサルタント (5.3%)、SE (5.3%)、学生 (5.3%) である。比較的時間の自由がある教員・研究者が非常に少なく、担当開発プロジェクトの進捗状態に参加が大きく左右されるソフトウェア技術者が大部分であることから、特定個人の継続的参加が期待できない半面、本論文で紹介したようなプログラム開発能力や、システム管理能力は非常に高い。

サブプロジェクト数は 18 であり、このうち翻訳サブプロジェクト数は 14 である。Jakarta プロジェクトのサブプロジェクト数は 19 であり、また Ja-Jakarta プロジェクトでは現在は ASF のトップレベルプロジェクトに昇格した Ant, Logging および Struts の 3 プロジェクトも対象にしていることから、一部の翻訳作業しか行っていないことが分かる。これは、翻訳作業を行う人数が絶対的に足りないこともあるが、開発者が実際に利用しているプロダクトは限られており、翻訳要求が出てこないプロダクトもあるからである。

翻訳作業に関しては、オフィシャルサーバの利用開始日である 2002 年 11 月 17 日から 2004 年 3 月末

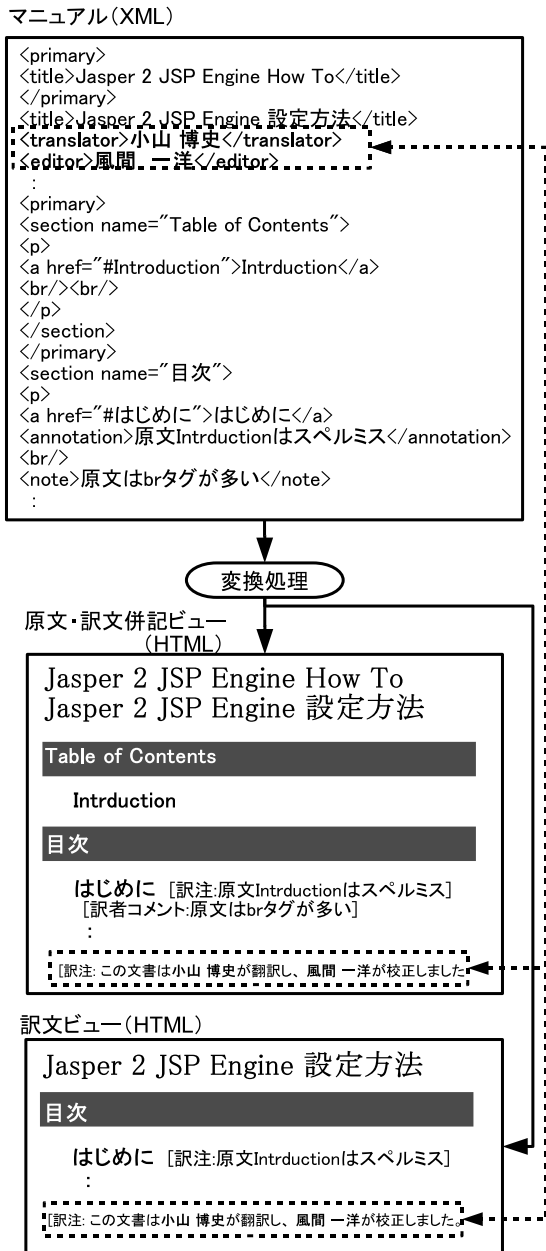


図 2 マニュアルドキュメントの変換例

Fig. 2 Example of transformation of manual documents.

表 2 メタデータの利用に関する投稿
Table 2 Contributions about the use of metadata.

| 番号 | 話題 | 投稿件数 | 参加人数 |
|-----|--------------|------|------|
| (1) | 翻訳進捗管理表生成ツール | 22 | 4 |
| (2) | 訳注タグ | 18 | 4 |
| (3) | 原文と訳文の併記 | 14 | 4 |
| (4) | API ドキュメント変換 | 8 | 5 |

までの約 16 カ月の間にリリースした翻訳ドキュメントは 7 つ、このうち翻訳したファイル総数は 290 個 (3,149,713 バイト) である。これ以外に、翻訳途中のファイルも多く公開している。

本システムによりどの程度翻訳作業が進むようになったかの傾向をみるために、オフィシャルサーバ利用前の約 16 カ月の間を対象として調べたところ、リリースされた翻訳ドキュメントは 3 であり、このうち翻訳したファイル総数は 105 個 (1,242,600 バイト) であった。時期により知名度や参加人数に差があるために、単純には比較できないが、オフィシャルサーバを利用するようになってからの方が、バイト数でみると約 2.5 倍の量を翻訳できていることが分かり、効果があったと判断してもよいと思われる。

7.2 創発性の実例

創発性について、メタデータ付与による翻訳スキームを実例として、生まれた経緯を分析する。

翻訳作業用の trans ML を分析したところ、投稿されたメール 2,747 通のうち、3.3% がこれに関する話題であった。これらの投稿期間は、2003 年 1 月から 12 月までの約 1 年間にわたって投稿されていることから、メタデータ付与による翻訳スキームは長い時間をかけて議論されてきたことが分かる。この内訳を表 2 に示す。

参加者の総数は 8 名であるが、すべての話題に参加していた翻訳者は 1 名、3 つの話題に参加していた翻訳者は 2 名であった。また、参加者の役割を分析すると、要望を出す人、仕様を提案する人、実装する人などに自然と分担されていた。これから、メタデータ付与による翻訳スキームはごく一部の人間によって統一的に管理されていたわけではなく、参加者の小規模な議論の集大成として自然発生的に生まれたことが分かる。

7.3 管理負荷の分散化の評価

管理負荷は各翻訳サブプロジェクトにおけるリリースマネージャを含む各コミッタの CVS リポジトリへのコミット数の割合により評価する。これをコミット率と呼ぶ。さらに、従来 100% だったリリースマネージャの負荷がどの程度コミッタへ分散されたかをみる

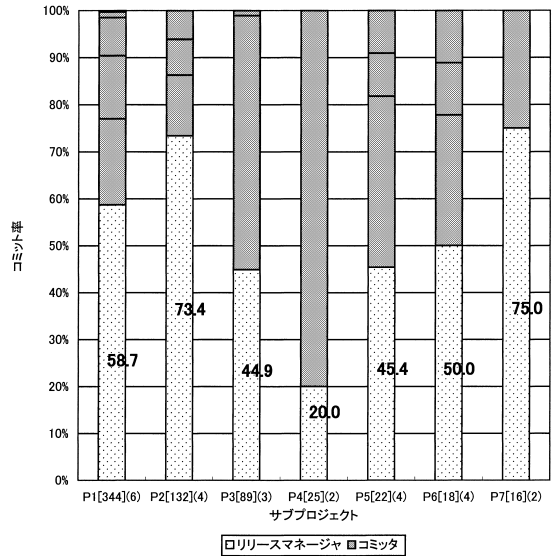


図 3 リリースマネージャとコミッタの CVS コミット率
Fig. 3 The CVS commit ratio of release managers and committers.

ために、リリースマネージャとコミッタの負荷比率を調べる。つまり、リリースマネージャのコミット率が $n\%$ であれば、従来の体制でリリースマネージャが行わなければならない仕事のうち $(100 - n)\%$ の仕事が他のコミッタに負荷分散されていることになる。

評価結果を図 3 に示す。縦軸はコミッタのコミット率を表す。ただし、リリースマネージャの率は一番下とし、数字を表示した。その上に同一プロジェクトに参加した各コミッタのコミット率を多い順に表示した。横軸の P1 から P7 は翻訳サブプロジェクトをコミット数の高い順に表示し、コミット数を [] 内に、コミッタ数を () 内に示した。リリースマネージャとコミッタは別の色で区別している。ただし、P1 のコミット率 0.3% のコミッタは、値が小さくグラフでは識別できない。

活動中の翻訳サブプロジェクトにおいて、平均コミッタ数は 3.6 人であることから、互いに協調して作業していることが分かる。また、コミッタの平均参加翻訳サブプロジェクト数を調べたところ 1.7 と比較的少なかった。これは各自が興味のあるサブプロジェクトを自ら担当した結果、自然と担当が分散したと考えられる。

リリースマネージャの管理負荷の平均は 52.5% であり、ほぼ半分減少した。ただし、他のコミッタと比べるとコミット率は高く、リリースマネージャが 1 番の翻訳サブプロジェクトの割合は 71.4% であった。これは、翻訳サブプロジェクトの翻訳進行管理や成果物

の公開というリリースマネージャだけが担当する作業が多いからであると推測できる。

P2 のリリースマネージャの負荷が高いのは、比較的時間に余裕があり、積極的に作業を進めたからである。特定の人に作業負荷が集中しても、このような場合は作業の停滞につながらず、問題はない。P7 もリリースマネージャの負荷が高いが少し事情が異なる。プロジェクトが翻訳を開始したばかりで、対象期間中リリースマネージャが事前準備を行っており、他のコミッタが作業を本格的に開始するに至っていないためである。P4 でリリースマネージャとコミッタで負荷の逆転が発生しているのは、時間が確保できなくなったリリースマネージャを他のコミッタが手助けしたからである。この1つの例だけでは断定できないが、動的に管理主体の委譲がされることもあるといえる。

ただし、P1 のように他に多くのコミッタが作業することでリリースマネージャの負荷が下がる場合や、P3 や P5 のように、ほぼ対等に作業している場合も観察されている。

以上の結果から、プロジェクトの管理負荷の分散化が実現できていると判断できる。

7.4 メタデータの有効性の評価

メタデータの有効性を評価するために、2003年10月から2004年3月の間に trans ML に投稿したコミッタおよび翻訳者を対象にアンケートを行った。メタデータの利用はサブプロジェクト単位で行っているため、利用経験がない翻訳者も対象とする。これは、メタデータの導入にともない若干の翻訳作業コストが増加する問題があるので、それに対する経験者と未経験者との反応に有意な差があるか知りたかったためである。以下の質問を著者を除く27名にしたところ、20件の回答があり、未回答項目が多い1件を除き19件を評価に使用した。

- (1) Ja-Jakarta プロジェクトでは、XML や Java ソースコードを翻訳するときに、翻訳者、校正者、訳注、コメントなどの情報を独自のタグを使って埋め込んで、校正用・配布用ファイル作成に使用しています。この方法は良いと思いますか？
- (2) 校正用と配布用と、異なるファイルを生成していることをどう思いますか？
- (3) 翻訳者と校正者の連絡用にコメント用のタグを設けていますが、それはどう思いますか？
- (4) これらの独自のタグを記入するのは面倒でしょうか？

集計にあたり、メタデータを利用した翻訳経験があ

表 3 アンケート結果

Table 3 Result of questionnaire.

| 質問 | グループ A | グループ B |
|-----|--------|--------|
| (1) | 4.5 | 3.5 |
| (2) | 4.2 | 3.7 |
| (3) | 4.1 | 3.8 |
| (4) | 4.0 | 3.2 |

るグループ A (13 件) と経験がないグループ B (6 件) に分け、得られた回答を質問 (1), (2), (3) は「非常に良い (5)」から「非常に悪い (1)」, 質問 (4) は「まったく面倒ではない (5)」から「非常に面倒 (1)」のように 5 段階評価した。この平均値を表 3 に示す。

一般的に比較的良好な評価が得られている。グループ A では、(1) の質問に対して 4 点以上の得点をつけた人は 100.0% であり、(2) から (4) の質問に対して 4 点以上の得点をつけた人は 84.6% であった。このことから、実際に作業した場合の満足度が高かったといえる。グループ B では、(1) と (4) の質問に対して「どちらでもない」(3 点) を選んだ人は 66.7% であり、(2) と (3) の質問に対して「どちらでもない」を選んだ人は 50.0% であった。未経験であるため、判断に困った結果の評価だと考えられる。それでも、すべての項目において平均が 3.0 以上であることから、未経験者からも比較的好意的に考えられているといえる。

なお、グループ A からは、メタデータ付与時に統合開発環境の入力支援機能やエディタのマクロ機能の利用についてのコメントも寄せられ、付与コストはさらに低減できると思われる。

以上の結果から、本システムでは人手によるメタデータの付与コストがかかっているが、翻訳者にとっても自分のクレジットの挿入や、より作業しやすい環境が得られるといった利点がある場合には、高く評価されることが分かる。

8. 今後の課題

現在、以下の 4 つの課題を検討している。

メタデータを、ビューの生成だけでなく翻訳進捗管理やバグ管理にも用いれば、さらに管理負荷の軽減を図ることができると考えられる。そこで、@status タグを用いた API リファレンスの翻訳進捗管理表の自動生成を試行中である。

訳語統一のためにサブプロジェクトごとに HTML 形式の訳語一覧表を用意しているが、この更新と閲覧に問題が生じている。そこで、訳語一覧表のデータ化と、それを用いた訳語表作成・更新の自動化を進めている。将来的には、辞書検索や機械翻訳用の辞書とし

でも提供する予定である。

英文・和文の併記，さらに多言語処理に関しては，`<primary>` タグは対応するテキストを DOM の木構造の異なるレベルに配置する点で，`@primary` タグは Java ソースコードを直接修正するために変更時の追従が難しくなる点で問題がある．これは以前のツールに問題が存在したからであるが，今後は XML の多言語処理のための仕様である `xml:lang` 属性を用いた記述や，翻訳したコメント部分を別ファイルで格納する機能を提供する javadoc コマンドの機能拡張モジュールである翻訳用ドックレットの利用などを検討したい。

頻繁に更新されるホームページやニュースなどの翻訳に関しては，マニュアルや API ドキュメントのような厳密性が求められないので，機械翻訳も検討している．そこで，協調型の機械翻訳システムである「訳してねっと」(<http://www.yakushite.net/>)²¹⁾ を公開前から試用している．たとえば，本プロジェクトの成果である訳文を対訳コーパスとして使用すれば，比較的妥当な品質の翻訳結果が得られる可能性があり，共同研究を継続する予定である。

9. ま と め

本論文では，オープンソースプロジェクトの分散協調翻訳支援システムの構築と運用について，分散協調作業におけるポジティブな効果の創発を促進する翻訳体制，メタデータの利用による翻訳支援という 2 つの視点を含めて述べた．さらに，活動状況や運用ログを分析した結果，管理負荷の分散化が実現できており，オープンソースモデルに基づいた翻訳体制がうまく機能していることが分かった．また，コミットおよび翻訳者へのアンケートを分析した結果，メタデータを利用した翻訳支援が，翻訳作業参加者に高く評価されていることが分かった．この結果から，分散協調翻訳の利点を引き出すような体制を構築できていることを実証するとともに，その問題点をある程度改善できたといえる。

本論文で述べた体制やシステムは，多くの人間の協調によって創発的に生み出されたものである．しかし，ML 参加人数が非常に多いにもかかわらず実作業に協力する人の割合が少ない，あるサブプロジェクトの作業は順調に進行するが別のサブプロジェクトの作業は停滞しがちななどのばらつきが観察されている．今後は，翻訳作業への協力をためらったり，サブプロジェクトの作業が停滞する原因をさらに解明するとともに，その原因をできるかぎり解消するようにシステムで支援することで，さらに活動を活性化したいと考えている。

謝辞 システム構築およびアンケートに協力していただいた Ja-Jakarta プロジェクトのボランティア，ネットワーク環境を提供していただいている (株) ガリレオ「訳してねっと」を提供していただいている沖電気工業 (株) の皆様に感謝する。

参 考 文 献

- 1) 青山幹雄：オープンソースソフトウェアの現状，情報処理，Vol.43, No.12, pp.1319-1324 (2002).
- 2) Lorge, I. and Solomon, H.: Two models of group behavior in the solution of eureka-type problems, *Psychometrika*, Vol.20, pp.139-148 (1955).
- 3) Laughlin, P.R. and Futoran, C.G.: Collective induction: Social combination and sequential transition, *Journal of personality and social psychology*, Vol.48, pp.608-613 (1985).
- 4) Okada, T. and Simon, H.: Collaborative discovery in a scientific domain, *Cognitive Science*, Vol.21, No.2, pp.109-146 (1997).
- 5) 植田一博，丹羽 清：研究・開発現場における協調活動の分析—「三人寄れば文殊の知恵」は本当か?，*認知科学*，Vol.3, No.4, pp.102-118 (1996).
- 6) Chi, M.: Conceptual change within and across ontological categories: Examples from learning and discovery in science, Giere, R. (Ed.), *Cognitive Models of Science: Minnesota Studies in the Philosophy of Science*, University of Minnesota Press (1992).
- 7) Raymond, E.: *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly (1999).
- 8) Berglund, E. and Priestley, M.: Open-source documentation: in search of user-driven, just-in-time writing, *Proc. 19th annual international conference on Computer documentation*, pp.132-141, ACM Press (2001).
- 9) 高橋邦夫：ネットワーク共同作業による教材作成，*理科教育メーリングリストシンポジウム* (1996).
- 10) W3C: Annotea Project. <http://www.w3.org/2001/Annotea/>
- 11) W3C: *XML Pointer Language (XPointer)*. <http://www.w3.org/TR/xptr/>
- 12) W3C: Resource Description Framework (RDF). <http://www.w3.org/RDF/>
- 13) 今村 誠，伊藤山彦，増塩智宏：Semantic Web の技術と応用の動向，*情報処理学会デジタル・ドキュメント研究会報告*，Vol.37, No.3, pp.17-24 (2003).
- 14) GNU: CVS—Concurrent Versions System. <http://www.gnu.org/software/cvs/>
- 15) 中村祐一：Apache プロジェクトに見るオーブ

ソース開発の実際, 情報処理, Vol.43, No.12, pp.1333-1335 (2002).

- 16) Apache Software Foundation: The Apache Ant Project. <http://ant.apache.org/>
- 17) W3C: Extensible Markup Language (XML). <http://www.w3.org/XML/>
- 18) W3C: XSL Transformations (XSLT) Version 1.0 (1999). <http://www.w3.org/TR/xslt>
- 19) Knuth, D.E.: *Literate programming (Center for the Study of Language and Information - Lecture Notes, No 27)*, Univ of Chicago Pr (1992).
- 20) Sun Microsystems, Inc.: *Javadoc*. <http://java.sun.com/j2se/1.4.2/docs/tooldocs/javadoc/index.html>
- 21) Murata, T., Kitamura, M., Fukui, T. and Sukehiro, T.: Implementation of collaborative translation environment: YakushiteNet, *Proc. 9th Machine Translation Summit System Presentation*, pp.479-482 (2003).

(平成 16 年 7 月 12 日受付)

(平成 17 年 2 月 1 日採録)



小山 博史 (正会員)

平成 5 年東北大学工学部情報工学科卒業。同年 (株) 日本電気入社。平成 8 年 (株) 日本電気退職。同年長野県職員。現在長野県工科短期大学校情報技術科講師。コンピュータ

と教育の研究に従事。電子通信情報学会会員。



風間 一洋 (正会員)

昭和 63 年京都大学大学院工学研究科精密工学専攻修士課程修了。同年日本電信電話 (株) 入社。現在、NTT 未来ねっと研究所主任研究員。分散協調処理, 情報検索の研究に従事。ソフトウェア科学会, ACM 各会員。



岡本 隆史

平成 9 年岡山大学大学院工学研究科博士課程前期修了。同年 (株) NTT データ入社。現在 (株) NTT データ技術開発本部所属。Web サービス, コンポーネントの研究に従事。



横田 健彦

平成 6 年東京工業大学大学院総合理工学研究科卒業。同年 (株) 東芝入社。現在 (株) 東芝研究開発センター知識メディアラボラトリー所属。ネットワークコミュニティベースの

知識共有システムの研究に従事。