

The Ja-Jakarta Project における分散システム構築と運用

小山 博史[†] 風間 一洋^{††} 岡本 隆史^{†††} 横田 健彦^{††††}

[†] 長野県工科短期大学校 情報技術科
^{††} 日本電信電話(株) 未来ねっと研究所
^{†††} (株)NTT データ 技術開発本部
^{††††} (株) 東芝 研究開発センター

E-mail: [†]koyama@cs.pit-nagano.ac.jp, ^{††}kazama@ingrid.org, ^{†††}okamoto@nttdata.co.jp,
^{††††}takehiko.yokota@toshiba.co.jp

あらまし 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おこなうボランティアプロジェクトである。全国に分散している貢献者達の協同作業を実現するために、主にオープンソースプロダクトを用いた分散システムを構築・運用し、同時に XML を利用した翻訳支援のためのサービスも提供している。本研究では、プロジェクトのメーリングリストの投稿、システムのログ及び開発者へのアンケートの解析結果に基づいてシステム設計や運用体制についての評価を行った結果、情報交換、情報公開、翻訳支援といったサービスが有効に活用されていることや、オープンソースモデルが翻訳作業に対しても有効であることを確認できた。

キーワード 分散システム, 翻訳支援, オープンソース, XML

Development and operations of the distributed system for the Ja-Jakarta Project

Hiroshi KOYAMA[†], Kazuhiro KAZAMA^{††}, Takashi OKAMOTO^{†††}, and Takehiko YOKOTA^{††††}

[†] Computer and Systems Engineering, Nagano Prefectural Institute of Technology
^{††} Network Innovation Laboratories, Nippon Telegraph and Telephone Corporation
^{†††} Research and Development Headquarters, NTT Data Corporation
^{††††} Corporate Research & Development Center Toshiba Corporation

E-mail: [†]koyama@cs.pit-nagano.ac.jp, ^{††}kazama@ingrid.org, ^{†††}okamoto@nttdata.co.jp,
^{††††}takehiko.yokota@toshiba.co.jp

Abstract The Ja-Jakarta project is a volunteer project to promote the japanization of open source products, and to translate their documents into Japanese. We have developed a distributed system constructed mainly with open source products and operates them to offer a cooperating work environment to the contributors scattered all over the country. We also offer a translation support service based on XML. We evaluated our system design and operation management by the analysis of the information in project mailing lists, system logs, and the questionnaire to our contributors. As the results, we confirmed that our services such as information exchange, information disclosure, and translation support have been effectively utilized. This paper thus reports that the open source model is effective for translation works.

Key words distributed system, translation assistance, open source, XML

1. はじめに

近年、オープンソースソフトウェアの注目度が高まることにより、企業開発者が業務の一環として開発に参加することや、企業の製品のオープンソース化が認められるようになり、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、企業の業務用システム構築に不可欠になりつつある [1].

日本でも、インターネットを介して影響を直接受けているが、多くのオープンソースプロジェクトは欧米が主流であり、母語が異なる日本では国際化作業や日本語化作業、及びドキュメン

トの翻訳を行わねばならない。そこで我々はオープンソースソフトウェアのドキュメント翻訳の分野のための分散作業支援システムを構築した。

本稿では、まず最初に日本独自の活動である Ja-Jakarta プロジェクトと、そのシステムの概要及び構築・運用により得た成果について述べ、さらにプロジェクトのメーリングリストの投稿、システムのログ及びアンケートを評価し、システムの有効性を示す。

2. Ja-Jakarta プロジェクト

Jakarta プロジェクトは、Java プログラム開発に関するオープンソースプロジェクトである。1999 年の JavaOne で、Sun Microsystems が Servlet API と JavaServer Pages のリファレンス実装を Apache Software Foundation に寄贈する発表があり、Jakarta プロジェクトを作成し、Tomcat という名称で開発を開始した。他にも、ビルドツール Ant や、フレームワーク Struts などの多くの実用的なプログラムを提供している。

Ja-Jakarta プロジェクトは、国内で独立に重複して行われていた Jakarta プロジェクトの日本固有の作業を統合して効率化するために、2000 年 9 月に発足したユーザ団体である。

Jakarta プロジェクトのプロダクトの日本語化・国際化、又はセキュリティ修正は共同で作業している。しかし、Jakarta プロジェクトのドキュメント及び API 仕様書の翻訳に関しては、日本独自の体制を構築する必要がある。

2.1 初期の体制の問題

初期の段階では、Jakarta プロジェクトのサブプロジェクトごとにドキュメントや API の翻訳サブプロジェクトを作成し、開発者（翻訳者）がメーリングリスト (ML) で情報交換しながら作業を行った。翻訳進捗状況や成果は、サブプロジェクトごとに選出されたまとめ役が一括管理し、企業の Web サーバにホームページを間借りして公開した。

しかし、プロジェクト規模が大きくなり作業量が増えるに従って、以下の理由から、作業期間の増大や情報の更新遅延などの点で深刻な問題が生じた。

- メールに頼った情報・データ交換は効率が悪く、情報共有が難しかった。
- ナビゲーションメニュー作成に Web サーバ独自機能を使用していたために、各作業者での検証が困難だった。
- 企業の Web サーバには外部から自由にアクセスできないために、Web サーバの管理は一人で行わなければならない、作業負荷が集中してしまった。
- スケジュール管理や校正作業を行う各翻訳プロジェクトのまとめ役の負荷が非常に高くなってしまった。

2.2 新しい体制への移行

そこで、情報交換の効率化とシステム管理・情報公開作業・翻訳作業の分散化を目標とした、次のような新しい体制への移行と分散作業支援システム構築が課題となった。

(1) プロジェクト独自のシステム運用方針の採用や、独自の翻訳支援システム開発・運用が可能になるように、オフィシャルサーバを設置した。

(2) オフィシャルサーバの安全性と分散化による効率性を両立させるために、Jakarta プロジェクトと同様なユーザ・開発者・コミッタ・管理者・PMC（プロジェクト運営委員）などの階層化された権限を設定し、貢献度に応じて強い権限を与えることにした。

(3) ユーザと開発者間の情報交換を促進させ、さらに情報公開を自動化することで管理コストを減少させるために、メール、掲示板、Wiki、Web、CVS を複合させたシステムを構築した。

(4) 複数の開発者による低負荷な翻訳作業を実現するために、CVS ベースの分散協調翻訳体制を確立し、それを支援するシステムを構築した。

一般的なオープンソースプロジェクトの開発体制 [2] と Apache Software Foundation の体制 [3] を参考にしながら、翻訳支援を中心とするシステムであることやプロジェクトに参加している開発者の人数などを考慮しながらシステムを構築した。

3. 分散作業支援システム

3.1 役割と責任分担

従来の開発者の役割と責任分担をより明確にするために、開発者、コミッタ、リリースマネージャ、管理者、PMC、PMC 議長の役割を設置した

コミッタは、十分な貢献を持つ開発者の中から、投票で選ばれる。コミッタはオフィシャルサーバの翻訳支援サービスを利用することができ、プロジェクトの将来に関する事項に関する投票権を得る。リリースマネージャは、作業計画を提案し、その管理を行う。PMC は、Ja-Jakarta プロジェクトの戦略的指針と成功に関して責任を負う。PMC 議長は、PMC の議事進行や対外折衝に責任を負う。

3.2 システム構成

ユーザ、開発者、管理者は従来と同様だが利用できるサービスが増える。このユースケース図を図 1 に示す。

掲示板を利用しやすくするために technical-bbs ML を追加したり、Anon CVS や cvsweb によるさらなる情報公開をしている。また、従来からあった jajakarta-general ML は一般的議論のための general ML と翻訳作業用の trans ML とに分けた。ユーザからのフィードバックを得るために従来からあった jajakarta-report ML は report ML と名前を短くした。イベントなどがあった場合にまとめられた情報を交換できるようにするために Wiki も使えるようにした。

システム構築にあたっては基本的にオープンソースソフトウェアを採用した。具体的には Debian GNU/Linux、Apache、Pukiwiki、Ant、Velocity、MySQL、Tomcat、Kvasir/Sora、Sen、CVS、cvsweb、poffix、fml nmap、iptables、OpenSSH、logcheck、logrotate、東風明朝/ゴシックフォントなどである。このうち、Content Management System の Kvasir/Sora や、日本語形態素解析器 Sen の開発者は Ja-Jakarta プロジェクトのシステム管理者でもあり、本システム向けの機能追加なども行って頂いている。また、アプリケーションによっては Java 開発ツールが必要であったため、Sun Microsystems J2SDK1.4.1

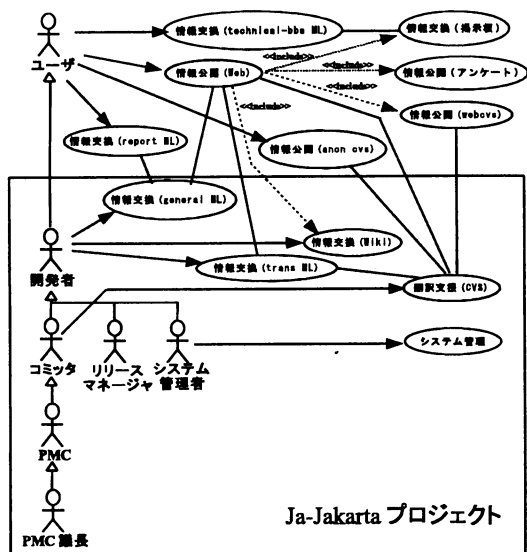


図 1 Ja-Jakarta プロジェクトのユースケース図
Fig. 1 The UseCase of the Ja-Jakarta project

を採用した。

3.3 情報交換

情報交換方法は ML と掲示板とに大きく分けられる。

3.3.1 メーリングリスト (ML)

本プロジェクトでは、一般のオープンソースプロジェクトと同様に、ML でユーザ、開発者間の情報交換を行う。発足当時は翻訳作業用 ML だけを提供していたが、コミュニティの成長につれ、Jakarta プロジェクトへのフィードバックや、JCP で策定している仕様、更には、Jakarta の体制の話など、翻訳以外のさまざまな話題が増加するという問題が生じた。

これに対しては、ML を翻訳作業とその他一般の話題とに分けて議論を行うことにより解決を図った。

3.3.2 掲示板

Jakarta プロジェクトへの注目が高まるにつれ、翻訳作業用の ML に翻訳作業以外のプロダクト使用法などの質問が目立ちはじめた。

そこで、ユーザサポートを行うための情報交換方法を検討した。一般的に、ユーザ間の情報交換には ML が良く用いられるが、ML を利用した場合、マニュアルを読まずに初歩的な投稿をユーザが行った場合に、本来の主旨が忘れられて ML におけるマナーの議論が行われることがあったり、使用しなくなったメールアドレスを ML から脱退させない人がいたりするため、システム管理者に負担がかかることが予想された。

これらの問題は、ML 自身がプッシュ型の情報発信を行うことと、自己責任で登録をして活用することに原因があると考え、本プロジェクトではプル型、ユーザ登録作業が不要な掲示板システムを採用することにした。利用者制限のない掲示板システムにすることにより、システム管理者の負担が増えないことと、興味が無い参加者に対する質問の配信をなくすことにより、参

加者間のコミュニケーション上のトラブルの軽減することとを図った。

このように掲示板システムを採用し運用をしていたが、掲示板をチェックしない開発者も多く存在したために、質問に気づかない、掲示板に投稿された注目すべき情報が見逃されている、といった意見がでてきた。そこで、掲示板に投稿された記事を配信するための配信専用 ML (technical-bbs ML) を新たに用意した。また、掲示板へ記事が投稿されるとこの ML へ自動的に配信される機能を掲示板システムへ追加した。ML の追加によりシステム管理者の負担は増えたが、これはシステム管理者を増やすことで対応した。

3.4 情報公開

翻訳したコンテンツは Web で公開している。当初、HTML 文書を Web サーバ上にアップロードして利用していたが、次のような課題があった。

(1) セキュリティ対策やインターネットの不正利用等を防ぐため Web へのアクセスを制限しているオフィスではコンテンツを見ることができない。

(2) XML 文書や Java のソースコードから HTML 文書を生成する場合、変換作業を毎回行ってから、コンテンツのアップロードを行う必要がある。また、新しいコンテンツ管理者が増えた場合、変換手順を修得する必要がある。

(1) については、Web で公開するだけでなく、翻訳が完成した段階で翻訳のアーカイブファイルもリリースし、ローカルのマシン上でもドキュメントを閲覧できるようにした。

(2) については、CVS と連係させ、XML や Java のソースコード等、原文書を CVS へコミットした時点で自動的に HTML を生成することにより効率化を図った。また、これらの処理はオフィシャルサーバ上に構築用ファイルやスクリプトを用意し、そこで実行できるようにした。

これにより、コミッタとなった開発者は変換手順を習得する必要がなくなった。また、必要であればサーバ上の構築ファイルやスクリプトファイルを自分のコンピュータ上へコピーして、確認のための変換を行ったり、変換手順修得のために調べたり、改良版を作成したりすることができる。

3.5 翻訳支援

3.5.1 翻訳体制

商業翻訳のように予算もなく、専任の翻訳者を確保できない本プロジェクトは、ネットワーク上の複数の開発者が協調しながら翻訳を進める分散協調翻訳体制を採用した。多忙な開発者の短い空き時間を利用して、1 つのファイルでもよいから翻訳して頂くように依頼する。個々の開発者の作業時間は短く、翻訳にも時間が掛かることが多くても、それを多くの人数で行うことで全体の翻訳速度の向上を狙う。

翻訳対象の文書は、CVS に保存する。各開発者は AnonCVS, cvsweb, Web サイトの翻訳管理ページ (CVS に保存されたデータから自動的に生成される) などから翻訳ドキュメントを自分のローカルコンピュータに複製して作業できる。ただし、CVS に翻訳ドキュメントを保存できるのは、コミッタだけである。さらに、コミッタの中から投票で選ばれたリリースマネージャ

により、スケジュール決定、翻訳ドキュメントのブロック化と優先度付け、作業分担、進捗管理などが行われる。

オープンソースプロジェクトの速い変更に従うために、翻訳対象ドキュメントの優先度の高いブロックから作業を開始し、翻訳が終了したドキュメントは随時公開されるので、読者は翻訳全体の終了を待つ必要はない。

3.5.2 分散協調翻訳体制の問題とその解決

ただし、従来の体制における経験から、分散協調翻訳体制には、次のような問題が存在することがわかっていた。

- 開発者に翻訳作業を理解してもらう時間が、参加の障壁になりやすい。
- 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。
- 頻繁な更新についていくためには、大胆な意識がしにくいので、直訳に近くなってしまう。
- 翻訳参加人数が多くなりがちで、かつ作業分担変更が多く、進捗管理が難しい。
- 翻訳サブプロジェクトごとに、局所的な規則ができやすく、全体の統一が難しい。

必ずしも翻訳の経験がない開発者に少ないドキュメントを翻訳してもらうだけの本方式では、少数の経験のある翻訳者を専任で割り当てて一気に処理できる商業翻訳ほどの高い翻訳品質や短い作業時間は期待できない。

しかし、オープンソースプロダクトのさまざまなドキュメントの日本語訳を公開する要求は高く、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。

そこで、本プロジェクトでは、これらの問題への対策手法を提案し適用した。以降では、実際の翻訳作業でどのような手法が適用されているかを説明する。

3.5.3 翻訳作業

翻訳対象のドキュメントは、大きくマニュアルと API リファレンスに分類される。Jakarta Project では、当初はマニュアルはプレーンテキストや HTML で記述されたが、現在は XML で記述されることが多い。XML ファイルは、Velocity の Anakia、Ant の XSLT タスク、Forrest などのツールを用いて HTML ファイルに変換し、ユーザに提供される。また、Java では、Donald Knuth が提唱した文芸的プログラミングの影響を受け、ソースコード中にコードだけでなく、その説明を独自のコメント形式を用いて同時に記述し、javac コマンドによるクラスファイル生成だけでなく、javadoc コマンドを用いて HTML 形式の API リファレンスを生成することができる。

つまり、本プロジェクトで翻訳対象となるのは、XML ファイル又は Java ソースコードである。両者に共通する特徴は、構造化ドキュメントであり、ドキュメントの変換過程が必要なことである。この特徴を利用し、次のことを実現した。

- 独自のタグを導入し、翻訳作業に必要なメタ情報をドキュメントと一緒に管理できるようにした。
- 翻訳規則をツールやスクリプト化し、その共通化と流用を促進して、独自の規則を減少させた。
- 翻訳の段階ごとに異なるビューを提供することを容易に

した。

ここでの「翻訳の段階ごとに異なるビュー」とは、翻訳の初期段階では原文と訳文の比較ができるように両方を含んだビューを提供し、最終校正の段階では訳文のみのビューを提供することを意味している。

たとえば、マニュアル (XML ファイル) の翻訳作業では、翻訳者、校正者、進捗状況、翻訳更新日を、それぞれ<translator>タグ、<editor>タグ、<status>タグ、<update>タグで記述し、翻訳者・校正者の明確化、翻訳文中のクレジット表示、進捗管理表の自動生成などに使用している。

また、和訳だけでなく、原文も<primary>タグで囲って残すことで、校正用の原文・和訳併記形式とリリース用の和訳みの形式などの異なるビューを生成できるようにしている。

さらに、訳者注を<annotation>タグで、訳者と校正者間の連絡事項を<note>タグで記述する。前者は最終の翻訳文に脚注として残るが、後者は訳者と校正者だけが見るもので、最終の翻訳文には残らないという違いがある。

API リファレンス (Java ソースコード) の翻訳作業でも、同様に@translator タグ、@editor タグ、@status タグ、@primary タグ、@annotation タグ、@note タグを使用している。

ただし、現在の手法では、複数の言語を同時に翻訳することはできない。これは、Sun Microsystems が API リファレンスの翻訳で行っているように、原文と訳文を異なるファイルとして保存すれば実現できるが、効率的に作業を進めるためには、専用でマルチプラットフォームの翻訳エディタが必要になると思われる。しかし、現在は XML ファイルの翻訳作業には各自好みのエディタを使っているために、エディタ上における作業利便性の方を優先している。

3.5.4 訳語管理

訳語統一のために、サブプロジェクトごとに訳語一覧を用意し、それに従って作業を行っている。現在は、さらにそれぞれの訳語一覧表のデータを統合し、訳語表作成・更新の自動化を進めている最中である。また、統合したデータは、辞書検索や機械翻訳用の辞書としても提供する予定である。

なお、本プロジェクトは、協調型の機械翻訳システムである「訳してねっと!」に初期の段階から関与し、一般公開前から数人の開発者がシステムを試用してきた [4]。翻訳の品質や本プロジェクトとのシステムとの連携などの課題があるものの、訳語統一を目的とした下訳作成や、Jakarta プロジェクトのニュースなどの翻訳の可能性を感じており、引き続き共同研究をしていく。たとえば、前述の手法で作成された翻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文が得られると考えている。

4. 評価

各サービスについて評価する。ただし、オフィシャルサーバ移行作業は、メーリングリストの移行、CVS の導入、掲示板の移行などのように段階を追って行われたため、ログ保存の開始時期が異なることに注意されたい。

4.1 情報交換

4.1.1 メーリングリスト (ML)

MLは、オフィシャルサーバ移行前から開発者間の情報交換サービスとして使用された。移行は2002年11月17日に行った。

メール投稿数の月毎の推移を、図2に示す。移行前の jakarta-general ML (2001/9/28-2002/11/17)、短期的なイベント用の javaone-bof-2002 ML (2002/09/10-2002/10/03)、移行後の general ML (2002/11/17-), trans ML (2002/11/17-), 短期的なイベント用の jtc-2004 ML (2003/12/25-) すべてを対象として含めている。

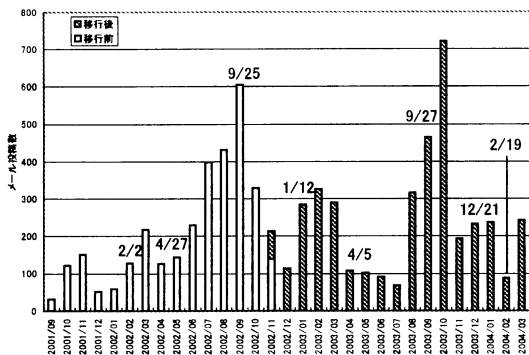


図2 メール投稿数の推移

Fig. 2 Changes in the number of incoming e-mails

jakarta-report ML, report MLについては、それぞれ jakarta-general ML, と general ML に同じものが投稿されるように設定していたので、実質的に含まれている。ただし、trans MLには CVSによって自動的に投稿されるメールもあり、それについては移行前に該当するものがないため、ここでは排除した。活動とミーティングなどのイベントとの相関関係を知るために、各イベントの開催日時も記入した。ほとんどのイベント後にメール投稿数が増加しているのがわかる。

次に、この推移より ML 移行の影響について述べる。2002年9月、10月はメール投稿数が非常に多い。これは、JavaOne2002のイベントと、オフィシャルサーバへの移行の議論が活発にされたためである。実際に11月にMLの移行が終了すると、イベントも特になく、年末時期に入ったため投稿数が減っている。しかし、翌年からは順調に推移していることから、移行はスムーズに行えたと言ってよい。実際に、1週間毎の平均投稿数は、移行前は51.8通で、移行後は56.4通となっており移行後の方が増えている。

4.1.2 掲示板

移行は2003年8月7日に行った。掲示板の記事投稿推移を図3に示す。移行直後はあまり変化はないが、2003年9月以降は従来の2倍以上の投稿数がある。

投稿数増加の要因としては、掲示板の新機能である technical-bbs ML への自動配信機能による効果、後述する Web サイト

アクセス数が増加した影響、general MLにて出されたユーザーインターフェースの改善要望対応の結果などが考えられる。

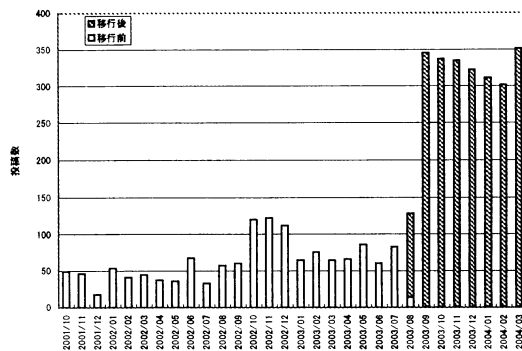


図3 掲示板投稿数の推移

Fig. 3 Changes in the number of the contributions to our BBS

4.2 情報公開

情報公開用の Web サイトアクセス数の推移を図4に示す。

2003年9月から急激にアクセス数が増加している。これは、掲示板は Web サーバ上で動作しているの、掲示板の投稿数が増えたことが原因のひとつと考えられる。

実際にアクセス数が多かった URL のトップ10について調べると、トップ10に含まれる掲示板へのアクセスは2003年8月においては53,000(全体の5.7%)程度であったのが、2003年9月においては190,000(全体の11.7%)程度に増加しており、影響が大きいと考えられる。ただし、Web サイトアクセス数増加により掲示板への投稿が増加したという相乗効果も考慮する必要がある。

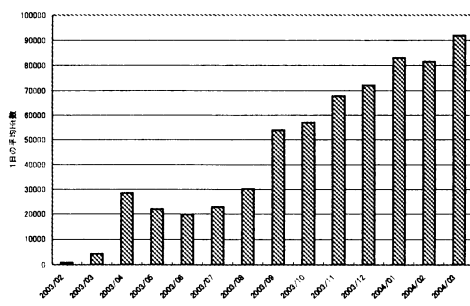


図4 Web サイトアクセス数の推移

Fig. 4 Changes in the number of the accesses to our website

この他の要因としては、Ja-Jakarta プロジェクトのメンバが雑誌などに記事を執筆する活動を通して Ja-Jakarta サイトの知名度が上がったこと、覚えやすい URL になったことや、複数のサイト管理者による頻繁な更新によりリピータが増えたことの影響が考えられる。

4.3 コミッタへのアンケート

コミッタを対象にして、オフィシャルサーバで提供しているサービスについてアンケート集計をした。回答者は20名中9名であった。

アンケートの各項目のうちサービスの必要性について表1に示す。質問事項によって回答の選択肢が異なるため、選択肢の内容によって「必要(いつも読んでいる)(10)、必要(たまに読む)(7)、必要(ほとんど読まない)(5)、いらぬ(0)」、「よく使っている(10)、使う(7)、使わない(3)、いらぬ(0)」のように得点対応させている。必要性は単純平均(総得点/回答者数)で算出している。

表1 サービスの必要性
Table 1 Necessity for the services

サービス名	必要性(10点満点)
general ML	10.0
report ML	7.9
CVSコミットをtrans MLへ流す	7.2
technical-bbs ML	9.4
CVS+Antによる翻訳状況のWeb更新	9.4
翻訳の段階ごとに異なるビュー	9.4
翻訳状態などを表す独自タグ	10.0
アンケートシステム	8.4
shell	8.7
cvswab	7.2
Anon CVS	5.4
掲示板	7.9
Wiki	8.5

どの項目も高い評価を得ている。trans MLとgeneral MLの2つにMLを分けたことは評価されている。翻訳支援に関係する各項目(CVS+Antによる翻訳状況のWeb更新、翻訳の段階ごとに異なるビュー、翻訳状態などを表す独自タグ)も非常に高い評価である。AnonCVSについては他と比較して低いが、これはコミッタであればSSH経由でCVSへ直接アクセスできるためだと考えられる。

貢献への影響についても調べた。

(1) どのようなときに貢献したいと思うか?(複数回答あり)

ミーティングの前後(2)、イベントの前後(2)、仕事が暇なとき(6)、その他(1)

(2) どのようなときに貢献できなくなると思うか?

ミーティングがないとき(0)、イベントがないとき(0)、仕事が忙しいとき(8)、その他(1)

(3) オフィシャルサーバの機能・性能は貢献する気持ちに影響するか

大きく影響する(0)、影響する(7)、どちらともいえない(1)、影響しない(1)、まったく影響しない(0)

貢献度に対しては、仕事による影響が大きかった。また、図2のMLの分析結果からは、イベント後にメールが増加しているが、アンケートの結果からはイベントの前後に貢献する気持ちが増加するコミッタが少ないことも判明した。イベント前後のメールの内容をより詳細にみていくと、イベント開

催に関する議論、イベントを通して実際に会った活動的なメンバーのコミッタへの推薦と投票、新プロジェクトの立ち上げに関する議論や投票等がメール数増加の原因であった。コミッタに対するアンケートの結果からはイベントと活動の活性化に対して有意な結果は得られなかったが、イベントの前後で投票等のコミッタの意志決定が必要な議論がなされることから、プロジェクトの活動に与える影響は大きいと言える。

オフィシャルサーバの機能・性能に対しては殆どの回答者が影響すると答えている。サービスの必要性でも高い得点がでていることからわかるが、貢献を促す環境のひとつとしてオフィシャルサーバで提供している機能は大きな意味を持っていることがわかった。

5. おわりに

ユーザが本プロジェクトに期待することは、妥当な品質の多数の日本語訳をすみやかに作成することである。この要求に対し、我々はネットワーク上の分散作業を支援するシステムを構築し、分散協調化・自動化を前提にした翻訳プロセス改善をおこなった。メール・掲示板・Webの利用履歴とアンケートの評価結果は、このようなオープンソースモデルが翻訳作業に対しても有効であることを示している。

ただし、オープンソース活動のような非集中型の活動の分析は、複数の要素が関係するために、詳細な分析が困難であることも判明した。今後は、システムに、より詳細に利用履歴を収集・分析する機能の組み込みを検討したい。

なお、このようなオープンソースモデルでは、各自のモチベーションが一番重要であり、我々の解析結果は、システムの使いやすさだけでなく、プロジェクトの施策もモチベーションの維持に密接に関係することを示している。たとえば、利用履歴の解析結果を、個々の開発者やリリースマネージャのスケジュール管理や、PMCのプロジェクト施策決定にフィードバックできれば、プロジェクトの活性化及びその維持に有効であると考えられる。そこで、人間という要素を考慮し、プロジェクト全体を活性化できるようなシステムの設計・構築手法の確立を目指したい。

6. 謝 辞

システム構築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頂いている(株)ガリレオ、「訳してねっと!」を提供して頂いている沖電気工業(株)の皆様へ感謝する。

文 献

- [1] 青山: “オープンソースソフトウェアの現状”, 情報処理, 43, 12, pp. 1319-1324 (2002).
- [2] 藤枝: “オープンソースソフトウェアの開発スタイルとその変遷”, 情報処理, 43, 12, pp. 1325-1328 (2002).
- [3] 中村: “Apacheプロジェクトに見るオープンソース開発の実際”, 情報処理, 43, 12, pp. 1333-1335 (2002).
- [4] T. Murata, M. Kitamura, T. Fukui and T. Sukehiro: “Implementation of collaborative translation environment: Yakushitenet”, the 9th Machine Translation Summit System Presentation, pp. 479-482 (2003).