

## 複数オープンソース間連携を要する インストールマニュアル自動生成法の提案

村上 幸一<sup>†‡</sup> 船曳 信生<sup>‡</sup> 徳永 秀和<sup>†</sup> 重田 和弘<sup>†</sup>

<sup>†</sup>高松工業高等専門学校 〒761-8058 香川県高松市勅使町 355

<sup>‡</sup>岡山大学 大学院自然科学研究科 〒700-8530 岡山市津島中 3-1-1

E-mail: <sup>†</sup>(murakami,tokunaga,shigeta)@takamatsu-nct.ac.jp, <sup>‡</sup>funabiki@cne.okayama-u.ac.jp

### あらまし

オープンソースソフトウェア（以下 OSS）では、そのソースコードをインターネットなどを通じて無償で公開している。そのため、設計の公開性、導入コストなどの面から、企業、官庁、大学などにおいて広く利用されつつある。しかし OSS では、開発者自身が本来の業務に加え、ドキュメント化やマニュアル作成を行う必要があることから、開発者の負担が大きく、それらの整備が進んでいない状況にある。特に、複数の OSS を連携させながら利用することも多く、OSS 利用の初心者にとっては、そのインストール作業でさえ容易ではない。そこで本研究グループでは、OSS 利用初心者のためのマニュアル整備の一環として、OSS インストールマニュアル自動生成法を提案している。本論文では、提案法に基づくシステムの実装を行うとともに、本提案法の複数 OSS 間連携を要する OSS のインストールマニュアルへの拡張を提案する。また複数の OSS 間連携を要する事例を用いた評価実験を行い、提案法の有用性の検証を行う。

キーワード オープンソースソフトウェア、マニュアル自動生成、パッケージ間連携

## A Proposal of an Automatic Manual Generation Method for Installing Several Open Source Soft Wares Cooperation

Yukikazu MURAKAMI<sup>†‡</sup> Nobuo FUNABIKI<sup>‡</sup> Hidekazu TOKUNAGA<sup>†</sup>  
and Kazuhiro SHIGETA<sup>†</sup>

<sup>†</sup>Takamatsu National College of Technology,

355 Chokushi-cho, Takamatsu-shi, Kagawa, 761-8058 Japan

<sup>‡</sup>Okayama University, Graduate School of Natural Science and Technology,

3-1-1, Tsushimanaka, Okayama-shi, Okayama, 700-8530 Japan

E-mail: <sup>†</sup>(murakami,tokunaga,shigeta)@takamatsu-nct.ac.jp, <sup>‡</sup>funabiki@cne.okayama-u.ac.jp

**Abstract** An open source software (OSS) is a software whose source code has been public through the Internet, so that everybody can use and modify it freely. As a result, a lot of companies, governments, and universities have now become using OSSs. However, the use of an OSS is actually hard for its beginners, because many OSSs have few good manuals, where an OSS has usually been developed by volunteers, and the installation of an OSS sometimes requires installations of several OSS packages before it. To solve this problem, we have proposed a method of automatically generating an installation manual of an OSS. In this paper, we implement the system for this method. Then, we extend our proposed method to generating the installation manual of an OSS that requires installations of other OSS packages at the same time. We conduct experiments using an OSS example requiring installations of four OSS packages to verify the usefulness of our method.

**Keyword** open source software, installation manual, automatic generation, package cooperation

## 1. まえがき

オープンソースソフトウェア（以下 OSS）とは、ソフトウェアのソースコードをインターネットなどを通じて無償で公開し、誰にでも、そのソースコードの改良、再配布を可能としたソフトウェアのことである。OSS は設計の公開性、導入コストのメリットから、現在、企業、官庁、大学などにおいて、システム開発、研究、教育など盛んに利用され始めている。

OSS では一方、市販ソフトウェアと異なり、一つの OSS 全体を一つの企業などの組織が責任を持って提供することがないため、ドキュメント化やマニュアルなどの不備が問題視されている[1]。マニュアル作成は、本来重要な業務であるが、OSS では、開発者自身が本来の OSS の開発業務に加えて、そのドキュメント化やマニュアル作成を行う必要があることから、開発者の負担が大きく、それらの整備が進み難い状況にある。特に、複数の OSS を連携させながら利用することも多く、その場合には、OSS 利用の初心者にとってはインストール作業でさえ容易ではない。

そこで本研究グループではこれまで、開発者の残したログ情報を元に、1 つの OSS のインストールマニュアル自動生成手法に関する提案を行ってきた[2]。本論文では、提案法に基づいたシステムの実装を行うとともに、本手法の複数 OSS 間連携を要する OSS のインストールマニュアル生成システムへの拡張を提案する。実装システムは、WEB システム上において、これまでに生成された OSS インストールマニュアルの共有やその再構築を行うことを前提としている。本論文では、既存の OSS インストールマニュアルを目的に応じて自由に組み合わせることのできる、複数 OSS 間連携インストールマニュアルの自動合成システムを提案する。また、複数の OSS 間連携を要する事例に対して、実装システムを用いてインストールマニュアルを作成し、その評価実験を行うことで、提案法の有用性の検証を行う。

ここで関連研究を簡単に紹介する。まず、香山らはログファイルや過去の操作履歴を学習に活用した事例として、操作履歴・学習履歴といったイベントデータを用いて、協調学習における学習資源共有と学習状態再生を提案している[3]。ここでは、後から参加した学習者が、先に学習を行っている他の学習者の履歴を参照することで、効率的に学習を進めることができることを意図している。しかし操作履歴・学習履歴収集のためには、香山らが提案する API をアプリケーション、教材作成の際に適用する必要がある。次に、映像ログを利用したマニュアル作成ソフトウェアとして、Screen Motion3 [4]、SOFT SIMULATOR [5] などが開発されている。しかしこれらは、映像ログ収集後に、手動での加筆・編集を前提としている。これでは、特に、複数の OSS パッケージ間連携を必要とするインストールマニュアルの作成時には、操作時間および操作ステップ数の増大から、映像ログ収集のためにかかる操作者への負担が大きいものとなる。

以下、本論文の構成を述べる。2.では提案手法について述べる。3.ではインストールマニュアル自動生成システムの実装について述べる。4.では実験条件と、結果について述べる。5.では本論文のまとめと、今後

の課題について述べる。

## 2. 提案手法

### 2.1. OSS インストールマニュアルの自動生成

図 1 に提案している OSS インストールマニュアル自動生成システムの概要を示す。本システムでは、熟練者による PC 上での OSS インストール時の操作ログをログファイルに記録し、それを用いてマニュアルを自動生成する。本システムは入力機能、編集機能、表示機能の 3 つの機能から構成されている。

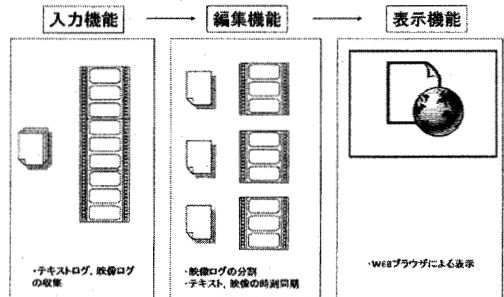


図 1 マニュアル自動生成システムの構成

#### ・入力機能

ターミナル上での入出力情報であるテキストデータの保存を行なうテキストログ機能と、スクリーンキャスティングソフトによる PC 上での操作を動画として保存する映像ログ機能を有する。ここで、テキストログはテキストファイルとして、映像ログは Flash ファイルおよび HTML ファイルとして保存される。

#### ・編集機能

保存されたテキストファイルを WEB 配信のための HTML 形式に変換するテキストマニュアル生成機能と、映像ログのシーン分割を行った上でテキストマニュアルとの関連付けを行う映像マニュアル生成機能を有する。

#### ・表示機能

WEB ブラウザ上に、テキストマニュアルおよび映像マニュアルの表示を行う。フレーム表示を用いて、ブラウザ画面の同一ページの上半分にはテキストマニュアル、下半分に映像マニュアルの表示を行う。また、タブブラウザの併用により、複数のマニュアルを表示することも可能としている。

### 2.2. 複数 OSS 間連携を要するインストールマニュアルへの拡張

WEB アプリケーション構築時の LAMP 構成に代表されるように、多くの OSS では、通常複数の OSS を組み合わせて使用されている。このような場合、各 OSS でのマニュアルに加え、複数 OSS を組み合わせたマニュアルを準備する必要がある。また、OSS には類似機能を持ちながら、実体として異なるものも多く存在する。そのため、連携の必要な OSS 毎にその類似製品も考慮した場合、マニュアルを整備すべき OSS の組み合わせは非常に大きなものとなる。しかしそれら全ての

組み合わせのマニュアルを作成することは困難である。また必要に応じて、その都度、必要なマニュアルを組み合わせで作成することも負担が大きい。

そこで本論文では、WEB システムを利用することで OSS を自由自在に組み合わせる、複数 OSS 関連インストールマニュアルの自動合成システムへの拡張を提案する。拡張システムの概念図を図 2 に示す。ここで、本拡張手法は、大学の研究室や企業での一部署といった小グループ内における知識共有を目的としている。

- ① グループ内において、個々の OSS インストールマニュアルのアップロードを行う。
- ② アップロードされたマニュアルは、WEB システム上のリポジトリに保存される。
- ③ 新規マニュアルの作成者は、システム構築のために必要な OSS を指示するシナリオファイルを作成する。
- ④ シナリオファイルに基づいて、リポジトリ内より既存マニュアルの合成が行われ、新規マニュアル作成者へ送られる。

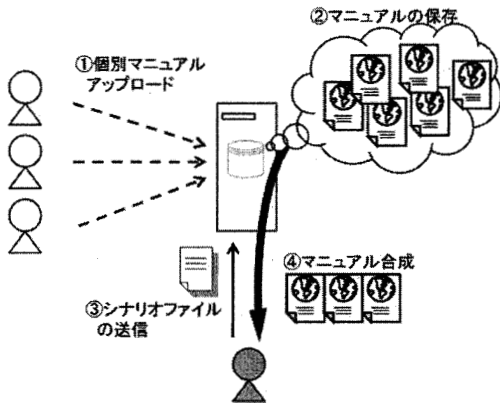


図 2 マニュアル自動合成システム概念図

### 3. 実装

#### 3.1. 入力機能

##### 3.1.1. テキストログの収集

ここでは Script コマンドを用いたテキストログ収集を行なう。Script コマンドは、アプリケーションのインストールなどのシェル上で行なった操作の入出力情報を記録するためのコマンドであり、操作ログをファイルに保存することができる。

今回は、`/etc/profile` に `script` コマンドの自動起動と `script` コマンドの `exit` にあわせて Shell を自動で `exit` されるように記載したシェルコマンドファイルを格納する。Shell 自身も自動で `exit` されるため、操作者に違和感をあたえずにログを保存することができる。また、`/tmp/log/ユーザ名/yyyymmdd` 上に、シェルの起動時刻をファイル名としたログが自動生成されるため、ログインした時間帯のログを適宜閲覧することにより、熟練者がどのような操作を行い、どのような結果が出力されたかを確認することができる。

##### 3.1.2. 映像ログの収集

映像ログを記録するための方法として、スクリーンキャスティングソフト[6]を利用する。これは、PC 上の操作を動画ファイルとして記録するものである。本論文ではその中で、OSS である“Pyvnc2swf”を用いる[7]。Pyvnc2swf は `vnc2swf.py` (録画ツール)、`edit.py` (編集ツール)、`play.py` (再生ツール) の 3 つの Python プログラムで構成されており、VNC プロトコルを経由して画面の動きを記録し、Shockwave Flash (SWF) ムービーを生成する。映像ログの収集においては、`vnc2swf.py` プログラムを使用する。`vnc2swf.py` には GUI (グラフィカル・インターフェイス) モードと CLI (コマンドライン・インターフェイス) モードの 2 つのモードが存在するが、本論文では自動実行の観点から CLI モードを使用する。

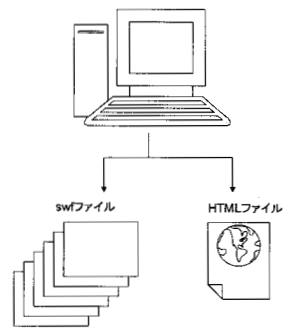


図 3 映像ログの保存

図 3 に示す様に、`vnc2swf.py` の実行では予め指定されたファイル名で `.swf` と `.html` の 2 つのファイルが生成される。`.html` ファイルには適切な HTML タグと、映像ログを操作するためのシークバーを生成するための javascript コードが含まれている。`vnc2swf.py` では画像をエンコードするために“`shape`”、“`video`”および“`vnc`”の 3 つの方式が選択可能である。今回生成速度が最も速く、`edit.py` コマンドによって SWF ムービーに変換することができる `vnc` を採用した。なおフレームレートとしてデフォルトの 12 フレーム/秒を用いて録画を行う。

#### 3.2. 編集機能

##### 3.2.1. テキストマニュアルの生成

Script コマンドで保存したログには、改行コードやバックスラッシュなどの制御コードも合わせて保存されているため、そのままでは非常に読みづらくなる。そこで、`sed` コマンドを用いることにより、これら制御コードを取り除くこととする。`sed` はストリームエディタであり、入力されたデータに対して、`s`(置換)、`i`(挿入)、`a`(追加)といったスクリプトで指定したフィルタリングを行うことができる。今回は、保存されたテキストログから制御コードを取り除くことにより可読性を高めると同時に、HTML タグを埋め込み、テキストログの HTML 化を行うものとしている。

### 3.2.2. 映像マニュアルのシーン分割

編集機能には Pyvnc2swf の edit.py プログラムを用いる。edit.py プログラムは、vnc2swf.py で生成したムービーファイルの編集、再構築を行うためのツールである。このプログラムでは .vnc ファイルを .swf ムービーに変換したり（形式変換機能）、ムービーから一部のフレームを取り出して別のムービーをつくる（ムービー分割機能）といった操作を行うことができる。

本システムでは、edit.py プログラムのムービー分割機能を拡張し、ターミナル上のコマンド入力時刻を元に、映像ログを分割する機能を実装している。

以下に処理の概要を示す。

- ① 録画が開始された時刻を記録する。
- ② 各コマンドが入力された時刻を記録する。
- ③ ①、②を元に、起動からコマンドおよび各コマンド間の経過時刻を計算する。
- ④ 経過時刻およびフレームレートを元に、分割コマンドに使用するスライド番号を算出する。
- ⑤ ④のスライド番号を元に、映像ログを分割する。

映像ログの分割例を図4に示す。

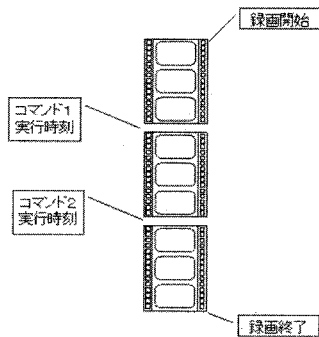


図4 映像ログの分割例

## 3.3. 表示機能

### 3.3.1. 個別マニュアルの表示

本システムにおいて生成されるマニュアルはオンラインマニュアルであり、WEBブラウザ上に表示する必要がある。またユーザーの利便性の観点からは、テキストマニュアルと映像マニュアルは、同時に表示されることが望ましいと考えられる。Internet Explorerなどの標準的ブラウザでそれらを異なるウィンドウ上に表示した場合、重なり合うことで、上側に表示されたウィンドウが下側に表示されたウィンドウの情報を隠してしまうという問題が生じる。

そこで、本システムのマニュアル表示には、上下2つに分割するフレームを用いるものとし、ウィンドウ上部にテキストマニュアル、ウィンドウ下部に映像マニュアルを表示するものとしている。テキストマニュアルの各コマンドには、映像マニュアルへのリンクが生成されており、リンクをクリックすることで、関連付けられた分割済みの映像マニュアルを表示できる。これによりユーザーは、テキストマニュアルで分かり

にくい箇所を、対応する映像マニュアルの参照により補うことが可能となる。また、映像マニュアルの操作のためにシークバーを採用する。ユーザーはシークバーをマウスでドラッグ、移動させることで、映像の早送り、巻き戻しを直感的に行うことができる。また、POUSE ボタンを利用することにより、映像を任意の位置で停止させることもできる。

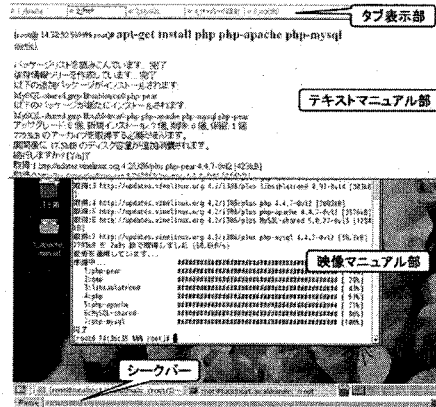


図5 個別マニュアルの表示例

### 3.3.2. 複数マニュアルの表示

複数 OSS パッケージ間連携を伴うインストール作業では、それぞれの OS のインストール作業を決められた順序通りに正しく行うことによってのみ、全体のインストール作業を成功させることができる。そのため、マニュアルにもユーザーが全体の手順を一目で見渡せるような工夫が必要となる。そのため本論文では、タブブラウザを利用して、それぞれの作業をタブで表示するものとしている。タブブラウザとは、タブを使うことで複数の WEB ページを切り替えて閲覧できる方式のブラウザの総称である。本システムでは、タブブラウザの中でも代表的な sleipnir[8]を用いる。

生成された個別 OSS マニュアルは、一つのタブで表示する。複数 OSS のマニュアルを表示する際は、これらを並べて表示することで、ユーザーは作業全体の作業数、および、現在操作を行っている作業の全体の中の位置づけを把握することができる。

## 4. 評価

複数の OSS 間連携を要する事例を用いた評価実験により、本システムを用いて自動生成したマニュアル（提案手法マニュアル）利用時と、従来のオンラインマニュアル（従来手法マニュアル）利用時での OSS インストール結果に関する比較を行なう。

### 4.1. 被験者

高松高専電気情報工学科4年生（大学1年生相当）20名を10名ずつの2グループに分け、一方に従来手法マニュアル、他方に提案手法マニュアルを用いた。

予備実験として、単一 OSS インストール作業での提案手法マニュアルと従来手法マニュアルの比較検証を行い、提案手法マニュアルに統計的な優位傾向が見られることを確認している [2]。本実験では、2グルー

ブ間の使用マニュアルを入れ替えを行うことで、利用マニュアルの変化による、課題終了者数の推移についても合わせて検証することとしている。

## 4.2. 実験条件

実験用 PC として、DELL 社 DIMENTION3100C (CPU CelronD 2.8GHz, メモリー 1 GHz) を用いた。また、OS 環境には、Windows XP Professional 上に、仮想 OS エミュレータである VMware Workstation をインストールし、VMware Workstation 上の Vine-Linux 4.2 上において、後述する課題を行なうこととした。

予備実験では、単一 OSS インストール作業に関する課題として、Tomcat のインストール、JSP の作成・実行を行った。また本実験においては、複数 OSS パッケージ間連携を要する実験課題として、Apache, PHP, MySQL, Xoops の 4 種類の OSS のインストールを順に行い、最後に各 OSS パッケージ間連携のための設定を行うものとした。Xoops は、コミュニケーションマネジメントシステム構築のための OSS である。[9]

## 4.3. 実験結果

### 4.3.1. 課題終了者数

#### ・予備実験

予備実験での課題終了者数を表 1 に示す。ここでは、提案手法マニュアルは従来手法マニュアルに比べ、課題終了者数が多い結果となった。また、直接確率検定では  $P=0.069$  となり、提案手法マニュアルには統計的な優位傾向が見られることがわかった。

表 1 予備実験における課題終了者数

条件	終了者数	未終了者数
提案手法	7	3
従来手法	2	8

#### ・本実験

本実験での課題終了者数を表 2 に示す。ここでも、提案手法マニュアルは従来手法マニュアルに比べ、課題終了者数が多い結果となった。しかし直接確率検定では  $P=0.17$  となり、提案手法マニュアルには統計的な優位差が見られないことがわかった。

表 2 本実験における課題終了者数

条件	終了者数	未終了者数
提案手法	6	4
従来手法	2	8

### 4.3.2. 実験間における課題終了者数の推移

2つの実験（予備実験、本実験）での課題終了結果の組み合わせごとに被験者数をまとめたグラフを図 6 に示す。図では、一行目を予備実験、二行目を本実験とし、提案手法マニュアルを“提案”，従来手法マニュアルを“従来”，と記載した。また、各実験における課題終了者を○，未終了者を×と表記した。図 6 より、予備実験で提案手法マニュアルにより課題終了+本実験で従来手法マニュアルにより課題未終了の被験者（提案○従来×），および、予備実験で従来手法マニュアルにより課題未終了+本実験で提案手法マニュアルにより課題終了の被験者（従来×提案○）の合計が最も多い。

これから、提案手法マニュアルの利用が課題終了に有効であると言える。また、マニュアルの違いによらず、課題を終了した被験者（提案○従来○，従来○提案○），マニュアルによらず課題未終了の被験者（提案×従来×，従来×提案×）も存在している。これらの被験者については、マニュアルよりも被験者本人の知識、技能の影響が大きいものと思われる。

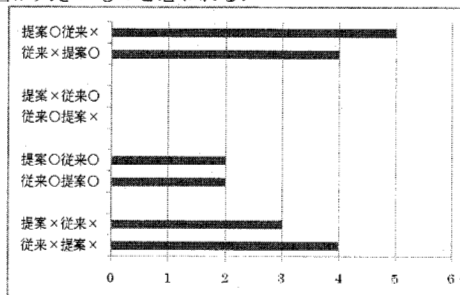


図 6 利用マニュアルによる課題終了者数の推移

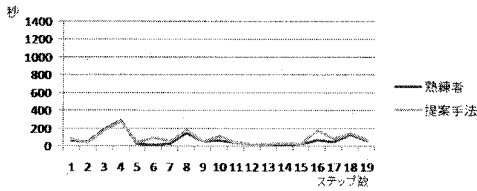
### 4.3.3. 被験者の行動分析

前節で、使用マニュアルの影響を大きく受けたと思われる被験者（提案○従来×，従来×提案○）に対し、NEM [10] を用いて評価を行うものとした。NEM とは、熟練者の操作時間と初心者の操作時間の効率比を表したものであり、ユーザビリティテストにおける問題点抽出手法として用いられる。ここでは、これらの実験で使用した提案手法マニュアルの作成者を熟練者とした。図 7 は、(a) 熟練者と従来手法マニュアルを用いた被験者の操作時間と比較したグラフである。グラフ横軸は操作ステップ数（ターミナル上でのコマンド入力数+GUI 上での入力数）を表している。また、グラフ縦軸はステップごとの操作時間を表す。図 7 (a)(b) より、提案手法マニュアルの利用者と熟練者の操作時間がほぼ同一となっているのに対し、従来手法マニュアルの利用者は熟練者の操作時間を大きく上回る操作ステップが多く見られる。また、図 7 (c) は、従来手法マニュアル利用者と提案手法マニュアル利用者の NE 比を表したグラフである。NE 比とは、熟練者と初心者の効率性の比を表したものであり、

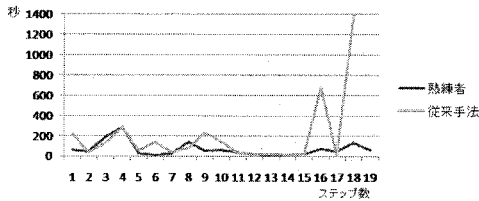
$$NE比 = \frac{\text{初心者ユーザの操作時間}}{\text{熟練者の操作時間}}$$

で定義される。本グラフより従来手法マニュアルに比べ、提案手法マニュアルでは NE 比が低く、有用性が高いと言える。従来手法マニュアルでは、ステップ 16, ステップ 18 において、NE 比が非常に高い結果となっている。被験者の操作ログを分析した結果、被験者がそれぞれでのエラー処理に長時間を要していることが明らかとなった。具体的には、ステップ 16 でのエラーはステップ 3 での設定ミス、ステップ 18 でのエラーはステップ 8 での設定ミスに起因することがわかった。これから、初心者にとって事前のステップにおける操作ミスを見つけ、その修正を行うことは非常に難しいと言える。これに比べ提案手法マニュアルは、初心者ユーザーの誤操作を防ぐマニュアルであったと言える。

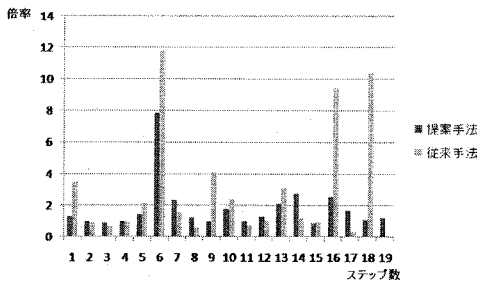
また、提案手法マニュアルでは、複数のマニュアルをタブにより切り替えて閲覧するため、タブの切り替え時の操作時間増大が懸念されたが、操作時間の増大は見られなかった。



(a) 操作時間の比較 (提案手法と熟練者)



(b) 操作時間の比較 (従来手法と熟練者)



(c) NE比 (提案手法および従来手法)

図7 NEM分析

#### 4.3.4. 被験者アンケート

表3 被験者アンケート項目

設問1.	マニュアルの中から必要な情報をすぐに探すことができた。
設問2.	マニュアルの情報はわかりやすかった。
設問3.	マニュアルから行っている内容の意味がわかった
設問4.	マニュアルから全体でどれくらいのステップ数(作業工程)があるかがわかった。
設問5.	マニュアルのすべての部分の通読はせずに、情報を拾い読みした。

表4 アンケート集計結果

条件	設問1	設問2	設問3	設問4	設問5
従来手法	3.2	3	2.7	3.3	3
提案手法	3.9	3.8	2.8	3.9	4.1

本実験終了後、被験者に対し表3のアンケートを行った。その結果、提案手法マニュアルでは、設問1「必要な情報をすぐに探すことができた」、設問2「通読せ

ずに、情報を拾い読みした」と答えた被験者が共に多く、従来手法マニュアルよりも必要な情報を見つけることが容易であったと言える。しかし、設問3「マニュアルから行っている内容の意味がわかった」と答えた被験者は両マニュアルともに少なく、OSSの知識、経験の少ない被験者にとっては意味を理解しやすいマニュアルではないことが明らかとなった。また、提案手法マニュアルにおいては、設問2「マニュアルの情報はわかりやすかった」、設問4「全体でどれくらいのステップ数があるかがわかった」と答えた被験者が多く、操作手順や作業の全体像を被験者に掴ませる上では有効であったと言える。

## 5. むすび

本論文では、1つのOSSのインストールマニュアル自動生成手法について実装を行うとともに、複数OSS間連携を要するOSSインストールマニュアルへの拡張について提案した。また、複数のOSS間連携を要する事例を用いた評価実験を行い、提案法の有用性の検証を行った。実験の結果、提案法によるマニュアルは、NE比の観点から有用性があることがわかった。今後の課題として、提案する拡張システムの実装が挙げられる。

## 文 献

- [1] OSS技術者のスキルセット調査  
<http://oss.mri.co.jp/ssoss/index.html>
- [2] 村上幸一, 船曳信生, 徳永秀和, 重田和弘, "ログファイルを利用したオープンソースソフトウェアのインストールマニュアル自動生成法の提案" 電子情報通信学会技術研究報告, ET2007-33, pp. 51-56, Sep. 2007.
- [3] 香山瑞恵, 岡本敏雄"協調学習での学習資源共有と学習状態再生," 信学論(D), vol.J88-D- I, no.1, pp.45-55, Jan.2005.
- [4] Screen Motion3  
<http://www.mswave.co.jp/>
- [5] SOFT SIMULATOR  
<http://www.patolis.co.jp/products/system/softsimulator/>
- [6] スクリーンキャスティングソフト -wikipedia-  
<http://ja.wikipedia.org/wiki/>
- [7] pyvnc2swf  
<http://www.unixuser.org/~euske/vnc2swf/pyvnc2swf-j.html>
- [8] sleipnir 公式サイト  
<http://www.fenrir.co.jp/sleipnir/>
- [9] XOOPSのインストール (LINUX 編)  
<http://kisa.homeip.net/wiki>
- [10] 黒須正明:『ユーザビリティテスト』. 共立出版, 東京, 2003