

操作ログを用い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

村上 幸一^{†1,†2} 船 曳 信 生^{†1} 徳 永 秀 和^{†3}
重 田 和 弘^{†2} 中 西 透^{†1}

Web 2.0 に代表される「柔らかなサービス」を提供するための基盤として、多くのオープンソースソフトウェア（以下 OSS）が開発、利用されている。OSS では、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おり、設計公開性、低い導入コスト、高いコード品質などの利点を有する。しかし OSS では、開発者自身がボランティア的にマニュアル作成を行う必要があることから、その整備が進んでいない状況にある。特にブログなど、「柔らかなサービス」を提供するための OSS では、複数の OSS 間を連携させながら利用することが多く、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。そこで本論文では、OSS 利用初心者のためのマニュアル整備を目的として、Web ベースの OSS インストールマニュアル自動生成法を提案する。提案法では、まず、熟練者による OSS インストール作業時の操作ログを用いて、テキストマニュアルと映像マニュアルで構成される、個別 OSS インストールマニュアルを生成する。次に、それらを組み合わせることで、複数 OSS 間連携を要する OSS でのインストールマニュアルを生成する。提案法を Linux 上に実装し、個別 OSS および複数 OSS 間連携を要する事例を用いた評価実験を通じて、その有用性を示す。

A Proposal of an Automatic Installation Manual Generation Method Using Operation Logs for Open Source Software

YUKIKAZU MURAKAMI,^{†1,†2} NOBUO FUNABIKI,^{†1}
HIDEKAZU TOKUNAGA,^{†3} KAZUHIRO SHIGETA^{†2}
and TORU NAKANISHI^{†1}

Recently, a variety of open source softwares (OSSes) have been developed and used as platforms of providing soft services in the Internet represented by Web 2.0. OSSes have advantages of the design openness, the low cost, and the high code quality. The source code of an OSS has been public through the Internet

so that everybody can use and modify it freely. However, most OSSes have few good manuals, because they have usually been developed by volunteers. As a result, even the installation of important OSSes including Blog is very hard for its beginners, particularly when the installation requires complete installations of multiple OSS packages at the same time. In this paper, we propose a method of automatically generating an installation manual of an OSS based on the Web technology, so as to provide the manuals to OSS beginners. Our method first generates the individual installation manual of an OSS composed of the text manual and the image manual, by using the operation set logs saved when a skilled person actually installed it. Then, it combines the set of manuals of the OSSes that are necessary to install an OSS. We have implemented the system based on our method on Linux and have conducted two experiments using installation manuals generated by this system. The comparisons with existing installation manuals show the superiority and the usefulness of our proposal.

1. はじめに

近年、インターネット上で、Web 2.0 に代表される、ユーザの手による情報の自由な整理や高品質な利用環境を実現する「柔らかなサービス」が注目されている。この「柔らかなサービス」を提供するための基盤として、多くのオープンソースソフトウェア（以下 OSS）が開発、利用されている。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、一定のルールの下で、誰にでも、ソースコードの改良、再配布を可能としている。そのため OSS は、設計の公開性、低い導入コスト、高いコード品質などの利点から、現在、企業、官庁、大学を含む多くの組織・機関において、システム開発、研究・教育活動などを目的に、広く利用されつつある。

一方、OSS では、市販ソフトウェアと異なり、1 つの OSS 全体を企業などの 1 つ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整備上の不備が問題視されている¹⁾。OSS では通常、開発者自身が本来の OSS 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ボランティアベースで行う必要があるため、開発者の負担が大

†1 岡山大学大学院自然科学研究科

Graduate School of Natural Science and Technology, Okayama University

†2 高松工業高等専門学校電気情報工学科

Department of Electrical and Computer Engineering, Takamatsu National College of Technology

†3 高松工業高等専門学校制御情報工学科

Department of Electro-Mechanical Systems Engineering, Takamatsu National College of Technology

大きく、それらの整備が進み難い状況にある。特に、ブログや Wiki, SNS などの「柔らかなサービス」を提供するための OSS では、複数の OSS 間を連携させながら利用することが多く、OSS 利用の初心者にとっては、インストール作業さえ容易ではない状況にある。

そこで本論文では、OSS 利用初心者へのマニュアル整備を目的として、操作ログからテキストマニュアルと映像マニュアルを自動生成する、Web ベースの OSS インストールマニュアルの自動生成法を提案する。ここで、“OSS 利用初心者”とは、「Windows 上で一般的なマウス・キーボードの操作が可能で、Microsoft Office に代表される市販ソフトの利用経験のある人で、Linux における CLI (コマンドライン・インタフェース) での操作経験や OSS の利用経験のない人」を意味する。また、多くの OSS は、これも OSS である Linux 上で開発されていることから、本論文では、Linux 上で開発された OSS を対象とする。

提案法では、まず、“OSS 熟練者”によるインストール作業時のログ情報を利用することで、個別 OSS のインストールマニュアルを自動生成する。ここで、OSS 熟練者とは、「OSS のインストール作業を間違わずに実行できる人」を意味する。本マニュアルでは、テキストマニュアルと映像マニュアルを併用することで、正確性と分かりやすさを実現する。次に、複数の個別 OSS のマニュアルを必要に応じて組み合わせることで、複数 OSS 間連携を要する OSS のインストールマニュアルを生成する。提案法に基づくシステムを実装し、「柔らかなサービス」に用いられる個別 OSS および複数 OSS 間連携を要する事例に対して、実際にインストールマニュアルを作成した。そのうえで、OSS の利用経験のない学生に対して、その評価実験を行い、提案法の有用性を示した。

以下、本論文の構成を述べる。2 章では、提案法の概要を述べる。3 章では提案法の実装について述べる。4 章ではマニュアル編集の評価実験を示す。5 章では、マニュアルのユーザビリティ評価について示す。6 章では、関連研究について述べる。7 章で本論文をまとめる。

2. 提案法の概要

本章では、提案法の概要を述べる。

2.1 個別 OSS インストールマニュアル生成

図 1 に本論文で提案する、個別 OSS のインストールマニュアル自動生成法の概要を示す。提案法では、OSS 熟練者による OSS インストール作業時の操作ログをファイルに保存し、それらを用いてマニュアルを自動生成する。そのため、提案法は、入力機能、編集機能、表示機能の 3 つの機能で構成される。

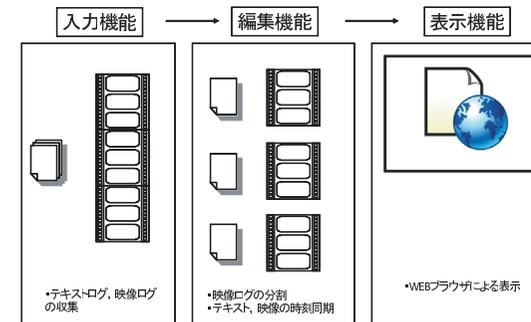


図 1 マニュアル自動生成法の概要

Fig. 1 Outline of automatic generation method of installation manual for OSS.

2.1.1 入力機能

入力機能では、OSS 熟練者による OSS インストール時の“操作ログ”の保存を行う。この操作ログは、OSS インストールマニュアル作成のために必要な操作データの記録であり、OS が恒常的に記録するコンピュータの利用状況やデータ通信の記録ではない。ここでは、ユーザ PC 画面内のターミナル上の入出力情報をテキストデータとして保存する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り PC 画面全体を動画として定周期で保存する映像ログ機能を有する。これらのログ収集は、熟練者による録画コマンドの実行により、テキストログ、映像ログの記録が連動した形で実行される。ここで、ログ収集中にインストール作業手順などで間違いが生じた場合には、これらの操作ログを破棄し、その取り直しを行う。そのため、基本的にはインストールに関係しないログは残さず、収集されたログはすべて利用される。なお、テキストログはテキストファイル、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。また、一連のインストール作業は、root 権限を有するユーザによる作業を想定している。

2.1.2 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編集機能と、映像ログのシーン分割を行ったうえで、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編集機能を有する。

2.1.3 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。その際、フレーム表示により、ブラウザ画面の上半分にテキストマニュアル、下半分に映像マニュアル

の表示を行うことで、マニュアルとしての利便性を高める。また、タブブラウザの併用により、複数のマニュアルをワンタッチで切り替えて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生成

一般に OSS においては、LAMP 構成や LAPP 構成²⁾ に代表されるように、複数の OSS パッケージを連携させてシステムを構築する事例が多い。また、2007 年に情報処理推進機構により報告された OSS 技術教育の現状とギャップ分析³⁾ において指摘されているように、OSS 教育上の問題点として、「OSS 製品間の組合せは相当数に達し変化も速いため、すべてを教えることは困難」である。実際、OSS には類似機能を持ちながら、実体としては異なるものも多く存在する。そのため、連携に必要な OSS ごとにその類似製品も考慮する場合、マニュアルを整備すべき OSS の組合せは非常に大きなものとなる。

そこで本論文では、あらかじめ作成しておいた個別 OSS インストールマニュアル群の中から、必要なものを自由に選択し、組み合わせることで、複数 OSS 間連携を要する場合のインストールマニュアルを提示することとする。そのためにユーザは、Web ブラウザ上において、OSS コアパッケージ名の入力後、表示される関連 OSS パッケージマニュアルの中から、実際に利用する OSS 群を選択する。また、利用する OSS パッケージ名を直接入力することで、マニュアル群を提示することも可能とする。

3. 提案法の実装

本章では、提案法の実装について、2.1 節の 3 つの機能ごとに述べる。

3.1 入力機能

3.1.1 テキストログの収集

“Script コマンド”を用いてテキストログ収集を行う。Script コマンドは、アプリケーションのインストールなど、シェル上で行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、操作ログをテキスト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、シェルを自動で `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。これにより、シェル自身も自動で `exit` されるため、操作者（熟練者）に負担を与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を自動生成する。これにより、ログインした時間帯のログを適宜閲覧することにより、操作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。

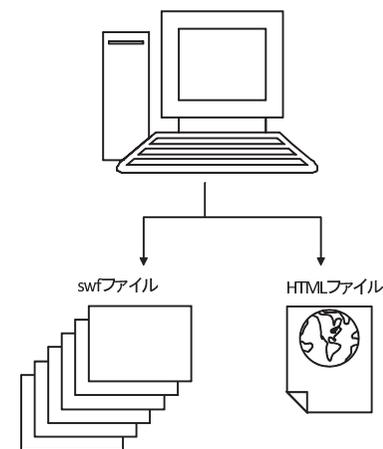


図 2 映像ログの保存
Fig. 2 Storage of image logs.

3.1.2 映像ログの収集

映像ログの記録のために、スクリーンキャスティングソフト⁴⁾ を利用する。これにより、PC 上の操作を動画ファイルとして記録する。本論文では、スクリーンキャスティングソフトとして OSS である “Pyvnc2swf” を用いる⁵⁾。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、ShockwaveFlash (SWF) ムービーを生成する。

映像ログの収集に利用する `vnc2swf.py` には、GUI (グラフィカル・ユーザ・インタフェース) モードと CLI (コマンドライン・インタフェース) モードの 2 つのモードが存在する。この中で、本論文では、自動実行の観点から CLI モードを使用する。図 2 に示すように、`vnc2swf.py` の実行ではあらかじ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。

また、`vnc2swf.py` では、画像をエンコードするために “shape”、“video”、“vnc” の 3 つの方式が選択可能である。今回、生成速度が最大で、`edit.py` コマンドにより SWF ムービーに変換することが可能な `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

マニュアルを“補助マニュアル”として位置づけている．映像ログをコマンド入力から次のコマンド入力直前までの最小作業単位でのシーン分割を行うことにより，それを実現している．また，このコマンド入力時刻でのシーン分割を行うことにより，コマンド入力後の PC による処理内容の映像をただちに確認することができる．

シーン分割には，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り，.vnc ファイルから.swf ムービーへの変換（形式変換機能），.swf ムービーから一部のフレームを取り出すことによる別ムービーの作成（ムービー分割機能），などの操作を行うことができる．今回，edit.py プログラムのムービー分割機能において，ターミナル上のコマンド入力時刻をもとに映像ログを分割する機能の拡張を行うことで，シーン分割機能を実装している．

以下に処理の概要を示す．

- ①録画開始時刻を記録する．
- ②各コマンドの入力時刻を記録する．
- ③①，②をもとに，録画開始から各コマンドの入力までの経過時間，および，各コマンド間の経過時間を計算する．
- ④経過時間およびフレームレートをもとに，分割コマンドに使用するフレーム番号を算出する．
- ⑤④のフレーム番号をもとに，映像ログを分割する．

映像ログの分割例を図 5 に示す．

ここで，動画の持つ動きの速さやその複雑性，情報量の多さのために，ユーザが動画の提示情報に追従できない，どこに注目しているのかわからない，といった問題が懸念される．既存研究として浦尾らは，工作や組み立てなどの学習領域を対象に，動画を用いた学習の有用性や認知負荷に関する検証を行っている^{6),7)}．その中で，動画の持つ動きの速さや複雑性に起因する学習効果の低下を，動画の速度調整，停止，再生，見直しなどの機能がもたらすシステムのインタラクティブ性により改善できることについて言及している．

そこで，本論文においてもインタラクティブ性向上のために，シークバーを採用することとした．ユーザ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，巻き戻しをインタラクティブ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可能である．

3.2.3 両マニュアル間のリンク

一般にテキストマニュアルは，映像マニュアルに比べ，通読性，正確性に優れている．こ

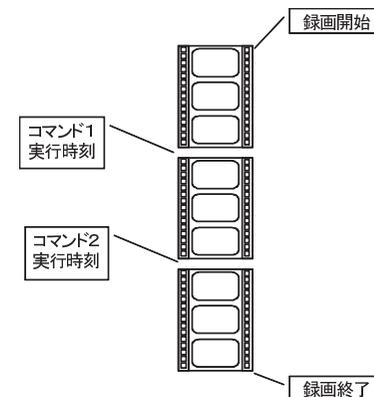


図 5 映像ログの分割例

Fig. 5 Example of division of image logs.

れに対し映像マニュアルは，熟練者による作業時の PC 画面の変化を完全に再現できるため，ユーザの本来の意図とは異なる解釈や誤操作を防ぐことができる．そこで，本論文では，その両者を必要に応じてユーザに同時に利用してもらうために，テキストマニュアル，映像マニュアル間の関連づけを行う．

通常ユーザは，テキストマニュアルを見ながら作業を進めることから，テキストマニュアル上の各コマンドに映像マニュアルへのリンクを生成する．これにより，ワンクリックで各入力コマンドに関連する，分割済みの映像マニュアルを表示可能とする．またユーザは，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．

3.3 表示機能

3.3.1 個別マニュアルの表示

提案法により生成される OSS インストールマニュアルは，オンラインマニュアルであり，Web ブラウザ上に表示される．その際，ユーザ利便性の観点から，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．しかしながら，Internet Explorer などの標準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，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い，同時には閲覧できないといった問題が生じる．

そこで，本論文のマニュアル表示では，ウィンドウ画面を上下 2 つに分割するフレームを

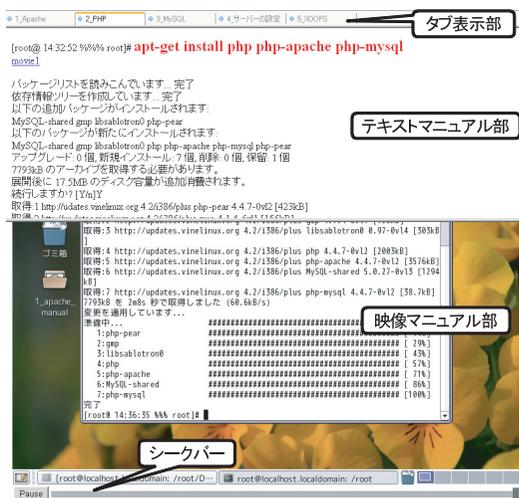


図 6 個別マニュアルの表示例

Fig. 6 Example of display of individual manuals.

用いる (図 6)。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ことで, 同時閲覧を可能とする。

3.3.2 複数マニュアルの表示

複数 OSS 間連携をともなう OSS のインストール作業では, 各 OSS のインストール作業を正しい順序で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は, ユーザが手順全体を一目で見渡せるような工夫が求められる。

この問題に対し本論文では, タブブラウザを利用し, 各 OSS のインストール作業をタブで表示することとしている。ここでタブブラウザとは, タブを使うことで複数の Web ページをワンタッチで切り替えて閲覧できる方式のブラウザの総称である。本論文では, タブブラウザの中でも代表的な Sleipnir⁸⁾ を用いる。各個別 OSS インストールマニュアルは, 1 つのタブで表示される。複数 OSS 間連携をともなう OSS インストールマニュアルを表示する際には, これらを並べて表示することで, ユーザは, 作業全体の中での現在の操作の位置づけを容易に把握することができる。

3.4 複数 OSS 間連携を要するインストールマニュアル生成システム

本論文の複数 OSS 間連携を要するインストールマニュアル生成システムは, Web アプリ

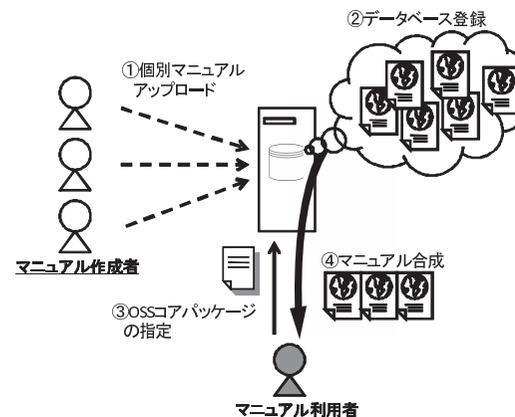


図 7 複数 OSS 間連携を要するマニュアルの自動合成手法の概念図

Fig. 7 Concept of automatic composition technique of installation manual for OSS requiring installations of multiple OSS packages to be integrated together.

ケーションとして, MySQL+PHP+Apache の組合せにより実装した。図 7 に本システムの概要を示す。また, 以下にその処理の概要を示す。

- ① マニュアル作成者が, 個別 OSS インストールマニュアルの作成を行い, Web サーバへアップロードする。
- ② マニュアル作成者が, 複数 OSS 間連携のための OSS パッケージ間の依存関係を, データベースに登録する。
- ③ マニュアル利用者が, ブラウザ上のフォーム画面において, インストールしたい OSS コアパッケージの指定を行う。
- ④ 提案システムが, Web サーバにおいて, OSS コアパッケージおよび, コアパッケージのインストールに必要な OSS パッケージインストールマニュアルの合成を行い, Web ブラウザに表示する。

ここで“マニュアル作成者”とは,「熟練者の残したログ情報を用いてマニュアルを作成する人」を意味する。また, 提案法では, 作成されたインストールマニュアルを Web システムを用いてアップロードする際に, マニュアルの作成日時, 作成環境 (Linux ディストリビューション, パーシジョン) OSS 情報 (パッケージ名, バージョン), を, マニュアル作成者が手入力する形でのメタデータづけを行っている。

表 1 OSS インストール環境の違い
Table 1 Difference of OSS installation environments.

1	Linux ディストリビューションの違い	リリースの公開時期が同じ リリースの公開時期が異なる
2	Linux のバージョンの違い	メジャーバージョンが異なる マイナーバージョンが異なる
3	OSS のバージョンの違い	メジャーバージョンが異なる マイナーバージョンが異なる

3.5 提案技術における制約

提案システムでは、マニュアル利用者とマニュアル作成時のインストール環境 (Linux ディストリビューション/バージョン, OSS バージョン) が同じ場合においてのみ、動作が保証される。そのため、特に複数 OSS 間連携を要するインストールマニュアルの生成には、多様なインストール環境下でのマニュアルを網羅することが重要となる。しかし、すべてのインストール環境における、すべてのバージョンを網羅することは非常に困難である。そこで、マニュアル利用者により指定されたインストール環境でのマニュアルが存在しない場合には、類似した環境で作成したマニュアルを、代替マニュアルとして表示することとした。

OSS のインストール環境が異なる場合としては、表 1 に示すように、1. Linux のディストリビューションの違い、2. 同一 Linux ディストリビューションにおけるバージョンの違い、3. OSS のバージョンの違い、の大きく 3 つが考えられる。これらのいずれかが違うだけでも、基本的には新たなマニュアル作成が必要となる。しかしながら、作成済みのマニュアルと、Linux ディストリビューションが違うがリリース公開時期がほぼ同じ場合、OS, OSS におけるマイナーバージョンのみが違う場合には、差異が小さい場合が多く、インストール作業に関しては、そのまま利用できる可能性も存在する。

そこで提案システムでは、マニュアル利用者が指定した Linux ディストリビューションでのマニュアルが存在しない場合には“リリース公開時期がほぼ同一の別ディストリビューションでのマニュアル”、同一 Linux ディストリビューションが存在する場合には“同一ディストリビューションで別マイナーバージョンでのマニュアル”を、代替マニュアルとして表示するようにしている。また、OSS のバージョンについても、指定バージョンのマニュアルが存在しない場合には、マイナーバージョンの異なるマニュアルを表示するようにしている。ここで、本システムで提示する代替マニュアルが、つねに正しく動作するかについては現状保証できていない。そこで、マニュアル上にその旨を明記するとともに、「正しく動作しない場合は、メールにてご連絡ください」と表示し、本システムを運用しながら、検証を進め

てゆくものとしている。

4. マニュアル編集性能評価実験

本章では、提案法によるマニュアル編集性能の高さを、評価実験により示す。

4.1 比較対象

本実験で比較対象とする、従来法でのマニュアル作成は、一般に以下の手順で実施される。

1. OSS インストール作業 (手動での画面キャプチャ)
2. マニュアル編集 (HTML の手動編集)
3. Web サーバへのアップロード

また、提案法によるマニュアル作成は、以下の手順により実施する。

1. ログ収集開始コマンド入力
2. 熟練者による OSS インストール作業
3. ログ収集終了コマンド入力 (編集開始コマンド自動実行)
4. マニュアル編集 (自動編集)
5. Web サーバへのアップロード

ここで、従来法における手順 1 と提案法における手順 1~3 については、ほぼ同等の作業内容である。また、Web サーバへのアップロードは、提案法、従来法とも共通作業である。そのため、作業時間に大きな差が生じるのは、「マニュアル編集」の部分となることから、両手法間でのマニュアル編集時間の比較を行うものとした。

4.2 被験者と実験条件

被験者として、ホームページの作成経験、および、本実験以前に OSS の十分なインストール経験のある、岡山大学の学生 5 名を被験者とした。実験課題は、動的な Web サイト構築のための OSS である Tomcat のインストール、JSP 作成、それらの実行に関するマニュアルの作成である⁹⁾。従来法では、HTML エディタ (Crescent Eve¹⁰⁾) を用いた手動編集を行うものとした。その際、できあがりの見本マニュアルを紙ベースで事前に配布することで、被験者間における手動編集マニュアルの等質性を確保することとした。また提案法ではプログラムによる自動編集とし、これら両手法における編集作業時間を測定した。なお、マニュアル作成に必要な操作ログならびに静止画像はあらかじめ準備しておき、被験者全員に同じものを利用させた。PC には、DELL INSPIRON 1501 (CPU Athlon × 2 1.7 GHz, メモリ 2 GHz) を用いた。

表 2 マニュアル編集時間の比較

Table 2 Comparison of edition time between two methods.

方法	提案法	従来法
平均作業時間	100.2 秒	59 分 14 秒

4.3 実験結果

表 2 に、本実験での作業時間の平均値の比較結果を示す。提案法は、従来法に比べ、非常に短時間でマニュアル作成が可能であることを確認した。ここで、表 2 の提案法のマニュアル編集時間は、PC による自動編集の処理時間である。提案法では、コマンド実行を除き、手動での作業を必要としない。そのため手動編集が必要な従来法に比べ、マニュアル作成の負担は大幅に軽減されることとなる。また、本実験における従来法での作業時間は、最短で 35 分 38 秒、最長で 75 分 10 秒であったことから、従来法では、マニュアル作成者個人の HTML の知識や文書作成の技量に大きく影響されるものと思われる。これに対し、提案法では、PC による自動編集のため、マニュアル作成者によらず、同一のマニュアルを同一の時間で作成することが可能となる。

5. マニュアルユーザビリティ評価実験

本章では、提案法により作成されるマニュアルの有効性を、個別 OSS インストールマニュアルでの実験（実験 1）、複数 OSS 間連携を要するインストールマニュアルでの実験（実験 2）を通じて示す。

5.1 実験 1

実験 1 では、個別 OSS インストールマニュアルとして、提案法を用いて自動生成したオンラインマニュアル（提案マニュアル）利用時と、従来のオンラインマニュアル（従来マニュアル）利用時における、OSS インストール結果に関する比較実験を行った。

5.1.1 被験者

高松工業高等専門学校電気情報工学科 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、一方を従来マニュアル⁹⁾、他方を提案マニュアルを用いるグループとした。その際、2 グループ間の被験者の等質性を確保するため、表 3 に示すアンケートにより実験課題の先行知識に関する調査を行った。各アンケート項目に該当した場合を 1 点として、全 10 問のアンケート結果の平均点がほぼ等しくなるように分けた。F 検定により得点分布が等分散ではないことが確認されたことから、等分散でない場合の t 検定を実施し、2 グループ間の被験者の先行知識に差のないことを確認した ($t(18) = 2.101, p = 1$)。

表 3 予備調査アンケート項目

Table 3 Questions in preliminary questionnaire.

1.	コマンドラインベース (UNIX, MS-DOS) での PC の操作は得意である。
2.	UNIX ベースの OS (LINUX, FreeBSD) を、授業以外で使用したことがある。
3.	プログラミングは得意な方だと思う。
4.	JAVA 言語を用いてプログラミングを行った経験がある。
5.	電化製品などは、操作マニュアルを読んでから、操作を行う方である。
6.	(PC やアプリケーションの使い方を調べるときは) 書籍よりもオンラインマニュアルを参考にすることの方が多い。
7.	自分の Web サイトを運営している。 (Web ログ, SNS を含まない)
8.	HTML, CSS などを用いてホームページを作成する機会が多い。
9.	自宅に自分用の PC を持っている。
10.	分からないことがあっても、あきらめない。

ここで、本アンケート結果の正当性を示すために、本実験終了後、課題終了に要した時間とアンケート得点間の相関を調べた。その結果、図 8 に示すように、右下がりの散布図となることが分かった。その相関係数を求めたところ、 -0.48 となり、アンケート得点と課題終了時刻との間には、中程度の負の相関関係¹¹⁾があることを確認した。

5.1.2 実験条件

実験用 PC として、DELL 社の DIMENTION3100C (CPU CeleronD 2.8 GHz, メモリ 1 GHz) を用いた。OS 環境として、Windows XP Professional 上に、仮想 OS エミュレータである VmPlayer をインストールし、VmPlayer 上の Vine-Linux 4.2 上において後述する課題を行うこととした。ここで予備実験で、ゲスト OS (Vine-Linux 4.2) 上にオンラインマニュアルを表示した場合、OSS 利用初心者が Linux における操作上の不慣れから、操作用のウィンドウが分からなくなってしまうといった事例が多く見られた。そこで Windows 上でマニュアルを表示しながら、VMware 上の Linux 上で操作するという形で両者の環境を分け、ウィンドウや背景色を変えることにより、操作上の混乱を極力排除できるように配慮した。

実験課題は、動的な Web サイト構築のための OSS である Tomcat のインストール、JSP 作成、それらの実行とした⁹⁾。本課題は、Linux、サーバ構築の知識がある者であればおおむね 10 分程度で終了可能である。そのため、課題の制限時間は、その 4 倍を見込んで 40

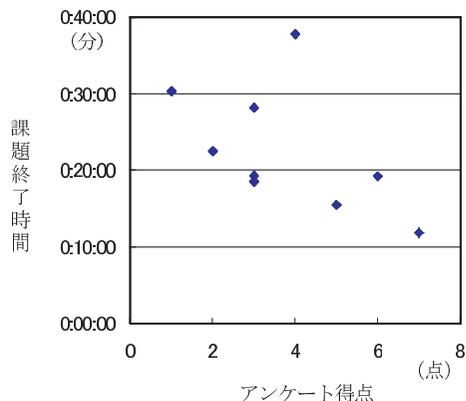


図 8 課題終了時間とアンケート得点の相関

Fig. 8 Correlation between task completion time and points in questionnaires.

表 4 実験 1 の課題終了者数の比較

Table 4 Comparison of numbers of students who completed tasks in Test No.1.

条件	終了者数	未終了者
提案マニュアル	7	3
従来マニュアル	2	8

分間とした。

5.1.3 実験結果

(1) 課題終了者数

2 種類のマニュアル間での課題終了者数の比較を表 4 に示す。表 4 では、従来マニュアルよりも提案マニュアルの方が終了者数が多くなっている。直接確率検定でも $p = 0.069$ となり、両マニュアル間の学習効果に統計的な有意傾向が見られることが確認できた。

(2) 課題の終了時間

課題終了時間の比較を表 5 に示す。従来マニュアルでは、20 分以内で終了するケースと、最後まで終了できないケースの両極端に分かれる結果となった。これに対し、提案マニュアルでは、11 分から 37 分までの間に連続的に終了時間が分布する形となった。

(3) 被験者アンケート

課題終了後のアンケートに、被験者から寄せられたコメントの例を表 6 に示す。従来マニュアルを利用した被験者からは、「Tomcat の動作確認のタイミングが分からなかった」、

表 5 実験 1 の課題終了時間の比較

Table 5 Comparison of task completion time in Test No.1.

提案マニュアル	従来マニュアル
11 分 50 秒	15 分 30 秒
18 分 29 秒	19 分 16 秒
19 分 14 秒	×
22 分 30 秒	×
28 分 9 秒	×
30 分 20 秒	×
37 分 47 秒	×
×	×
×	×
×	×

表 6 被験者アンケートのコメント

Table 6 Comments of examinees in questionnaires.

従来マニュアル

- ・Tomcat の動作確認のタイミングが分からなかった。
- ・JSP アドレスの指定方法が分からなかった。
- ・説明ばかりで、不手際が起きたときに何をしたらよいか分からない。
- ・できない理由、分からない理由がよく分からない。

提案マニュアル

- ・動画のスピードがゆっくりで分かりにくい。
- ・動画がどこで終わるか分からないから、何回もクリックするはめになった。
- ・やっていることがよく分からない。

「JSP アドレスの指定法が分からなかった」といったコメントが出され、被験者にとり、従来マニュアルから操作方法を読み取ることは容易ではなかったことがうかがわれる。一方、提案マニュアルを利用した被験者からは、映像マニュアル部の操作性に関する不満は寄せられたものの、「操作方法が分からなかった」といったコメントは見られなかった。

また両者ともに、「やっていることがよく分からない」というコメントが寄せられていたものの、提案マニュアルにおいては、そのような被験者でも課題を終了しているケースが見られることも分かった。

5.1.4 考 察

本評価実験の結果、まず、提案マニュアルは、従来マニュアルに比べてマニュアルとしての学習効果の面で統計的な有意傾向が見られることが分かった。また、表 5 より、提案マニュアルでは、22 分 30 秒以内に課題が終了した組と、それ以上費やして課題が終了した組の 2 つが存在するといえる。その理由として、前者の組では、主にテキストマニュアルのみを参照することで短時間で終了させているのに対し、後者の組は、映像マニュアルを多く参照したために時間を要したものと思われる。従来マニュアルでは、20 分以上要して課題を終了した被験者が存在しないことから、後者の組については、従来マニュアルであれば課題を終了することができなかったものと推定される。

5.2 実 験 2

実験 2 では、複数 OSS 間連携を要するインストールマニュアルに関して、実験 1 と同様の実験を行った。

5.2.1 被 験 者

実験 1 での被験者をそのグループを入れ替えて実施した。これにより、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検証することとした。

5.2.2 実 験 条 件

実験用 PC は実験 1 と同じである。複数 OSS 間連携を要する課題として、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。ここで XOOPS は、ブログなどに代表されるコンテンツマネージメントシステム構築のための OSS である¹²⁾。

5.2.3 実 験 結 果

(1) 課題終了者数

本実験での課題終了者数を表 7 に示す。本実験でも、提案マニュアルは従来マニュアルより、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、統計的な優位差は見られないことが分かった。そこで、2 つの実験結果間の関係を分析することとした。

(2) 2 つの実験間での課題終了者数の推移

2 つの実験（実験 1、実験 2）での課題終了結果の組合せごとに被験者数をまとめたグラフを図 9 に示す。図中 1 行目を実験 1、2 行目を実験 2、提案マニュアルを“提案”、従来マニュアルを“従来”、各実験における課題終了者を“○”，未終了者を“×”と表記した。

図 9 では、「実験 1 で提案マニュアルにより課題終了 + 実験 2 で従来マニュアルにより課

表 7 実験 2 の課題終了者数の比較

Table 7 Comparison of numbers of students who completed tasks in Test No.2.

条件	終了者数	未終了者
提案マニュアル	6	4
従来マニュアル	2	8

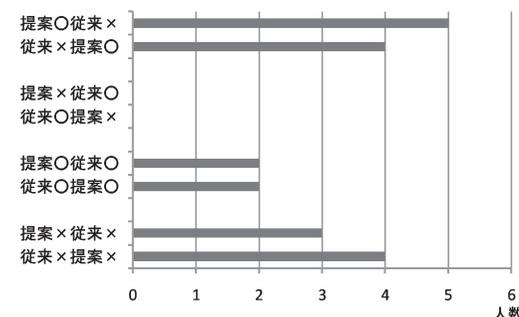


図 9 利用マニュアルによる課題終了者数の推移

Fig. 9 Change of number of students who completed tasks depending on used manuals.

題未終了」の被験者（提案 従来×）、「実験 1 で従来マニュアルにより課題未終了 + 実験 2 で提案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。これより、提案マニュアルの利用が課題終了に有効であるといえる。

(3) 被験者の行動分析

(2) 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^{13),14)}を用いて評価を行うこととした。NEM 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、NEM での熟練者を、提案マニュアル作成時の OSS 熟練者（= 第 1 著者）としている。OSS インストール作業時間は、同一 PC であれば、ほぼ同一となることから、今回は第 1 著者による 1 回の実験データを熟練者の操作時間とした。

図 10 は、熟練者と従来マニュアルを用いた被験者、図 11 は、熟練者と提案マニュアルを用いた被験者の操作時間を比較したグラフである。図中の値は、使用マニュアルの差異の影響を大きく受けたと思われる被験者（図 9 の“提案 従来×”、“従来×提案 ○”）の平均操作時間である。図 10 では、“提案 従来×”の被験者 5 名、図 11 では“提案×従来 ○”の被験者 4 名の操作時間の平均値を用いている。グラフ横軸は操作ステップ数（ターミナル

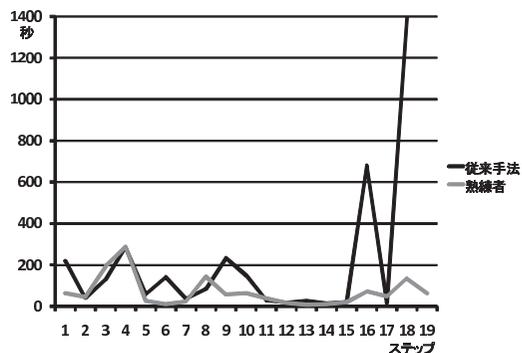


図 10 操作時間の比較 (従来法と熟練者)

Fig. 10 Comparison of operation time (between conventional manual and expert user).

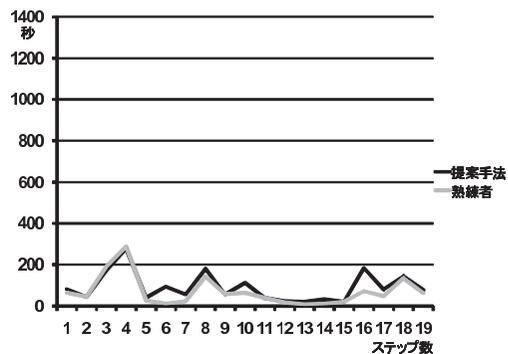


図 11 操作時間の比較 (提案法と熟練者)

Fig. 11 Comparison of operation time (between proposed manual and expert user).

上でのコマンド入力数 + GUI 上での入力数), 縦軸はステップごとの操作時間を表す。ここでは, 提案マニュアル利用者と熟練者の操作時間がほぼ同じであるのに対し, 従来マニュアル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。

図 12 は, 従来マニュアル利用者と提案マニュアル利用者の NE 比を表したグラフである。NE 比は, 熟練者と初心者の効率性の比を表したものであり, 次式で定義される。

$$NE \text{ 比} = \frac{TN}{TE}$$

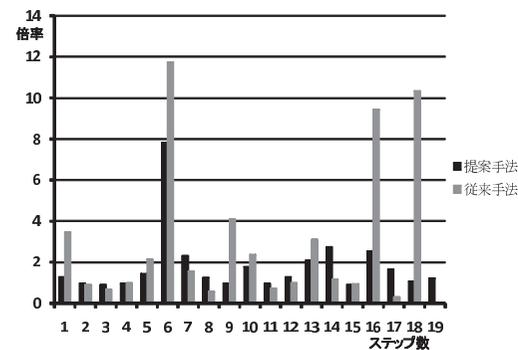


図 12 NE 比の比較

Fig. 12 Comparison of NE ratios.

TN: 初心者ユーザの操作時間

TE: 熟練者の操作時間

本グラフより, 従来マニュアルに比べ提案マニュアルの NE 比が低く, 有用性が高いといえる。従来マニュアルでは, 特にステップ 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。被験者の操作ログでは, ここでは, 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。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いえる。これに比べ提案マニュアルは, 初心者の誤操作を防ぐマニュアルであったといえる。なお提案マニュアルでは, 複数のマニュアルを切り替えて閲覧することによる操作時間の増加は見られなかった。

(4) 被験者アンケート

本実験終了後, 被験者に対し, 表 8 のアンケートを行った。そのアンケート結果を表 9 に示す。この結果から, 提案マニュアルでは, 設問 1「必要な情報をすぐに探すことができた」, 設問 5「通読せずに, 情報を拾い読みした」と答えた被験者が多く, 従来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いえる。しかし, 設問 3「マニュアルから行っている内容の意味が分かった」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。最後に, 提案マニュアルにおいては, 設問 2「マニュアルの情報は分かりやす

表 8 被験者アンケート項目

Table 8 Questions to examinees in questionnaire.

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分かりやすかった.
設問 3.	マニュアルから行っている内容の意味が分かった.
設問 4.	マニュアルから全体でどれくらいのステップ数が(作業工程)があるかが分かった.
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 9 アンケート集計結果

Table 9 Result to questionnaire in Table 8.

条件	設問 1	設問 2	設問 3	設問 4	設問 5
提案法	3.9	3.8	2.8	3.9	4.1
従来法	3.2	3	2.7	3.3	3

提案法：提案マニュアル，従来法：従来マニュアル

5：良い，4：やや良い，3：普通，2：やや悪い，1：悪い

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分かった」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うえでは有効であったといえる。

6. 関連研究

本章では，関連研究について述べる。

Linux 上における操作ログを用いた教育支援システムとして，川田らは，教師の PC 操作を監視し，キーボードやマウスのどのキー，ボタンが押されたかを自動的に表示する，X ウィンドシステムを用いたものを提案している¹⁵⁾。しかし，このシステムは，どのキー，ボタンが押されたかを単純に表示するシステムであり，操作対象に応じた注釈や操作説明の表示など，操作マニュアル生成に関する検討は行われていない。

操作マニュアル自動生成の研究として，入部らは，IT 講習会や情報リテラシ教育での使用を目的に，初心者向けの対話型ソフト学習システムを提案している¹⁶⁾。本システムは，教材作成システム，操作学習システムの 2 つから構成される。教材作成システムでは，教材作成者が対象とするソフトで一連の操作を実行することにより，自動的に操作イベント情報を取得し，操作説明用画像を生成する。しかしながら，本システムは，Microsoft 社が提供す

る MSAA (Microsoft Active Accessibility) を用いて開発されており，Linux 上で利用されることの多い OSS への適用は困難である。加えて，既存ログや既存マニュアルの再利用なども考慮されていない。

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，香山らは，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に関する提案を行っている¹⁷⁾。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られることを意図している。しかし各履歴の収集には，ソフトウェアそのものの改変が必要となることから，既存のソフトウェアや OSS への適用は難しく，標準化のための提案としての側面が大きかったと考えられる。

同じく香山らは，知識共有を目的とした協調プラットフォームを提案している¹⁸⁾。これは，アプリケーションウィンドウのイメージの共有という形で，メタレベルでのアプリケーション操作の共有を実現するものであり，Windows，Mac，Linux の 3 種類の OS 上で利用することができる。しかし，ここで共有されるのは，アプリケーション操作後の成果物であり，インストール作業などのアプリケーション操作ではない。そのため OSS インストールマニュアルの作成には利用できない。

動画や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 Motion3¹⁹⁾，SOFT SIMULATOR²⁰⁾，ViewletBuilder5²¹⁾などが発売されている。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ため，インストールマニュアル作成時の負担が大きいものとなる。特に，複数 OSS 間連携を要する場合，操作時間，操作ステップ数の増大から非常に大きな問題となる。さらには，これらのソフトウェアはいずれも，Windows 上での利用に限定された製品であり，Linux 上で利用されることの多い OSS のマニュアル作成への適用は難しい。

ここで，提案法の機能的な妥当性を検証するために，以上の関連研究の中で，代表的なマニュアル作成支援ソフトである ViewletBuilder5²¹⁾，マニュアル作成への応用が可能な入部らの研究¹⁶⁾を選定し，提案法との機能比較を行う。比較項目には，マニュアル作成の自動化度，マニュアルでのユーザ支援機能(操作支援，参照支援，理解支援)^{22),23)}，Linux 環境対応，複数 OSS 間連携のためのマニュアル合成機能とする。表 10 に比較結果を示す。これより，「生成マニュアルのアップロード自動化」，「操作支援でのエラー警告」，「理解支援での操作意味の解説」以外，提案法は各機能を充足しており，比較法よりも全体として優れているといえる。特に提案法は，Linux 環境に対応していること，マニュアル合成機能を有していること，に特徴があるといえる。

表 10 関連研究との機能比較

Table 10 Comparison of functions between two studies and proposal.

比較項目		Viewlet	入部	提案
自動化度	編集作業			
	アップロード	×	×	×
操作支援	文書 + 映像相互補完		×	
	エラー警告	×		×
参照支援	重要部分の強調			
	タイトル・目次		×	
理解支援	情報の区別化			
	操作意味の解説			×
Linux 環境への対応		×	×	
マニュアル合成機能		×	×	

ここで提案法において、「アップロード自動化」は、テキストログから、OSS パッケージ名、バージョン情報を抜き出す形での自動化を現在検討している。「エラー警告」は現在開発中であり、その試作を一部終了している。「操作意味の解説」は、利用する操作ログの中に操作意味の解説情報が含まれないことから、その自動化は困難であり、今後の課題である。

7. む す び

本論文では、「柔らかなサービス」を提供するための基盤整備を目的として、OSS 熟練者による作業時のログ情報を利用した、OSS インストールマニュアルの自動生成法を提案した。また、提案法を Linux 上に実装し、個別 OSS インストールマニュアルおよび複数 OSS 間連携を要するインストールマニュアルに対する評価実験を行うことで、提案法の有用性を示した。今後の課題として、関連研究で示した課題に加え、マニュアルとしてのユーザインタフェースの改善、提案法による様々な OSS のインストールマニュアルの作成、それらの評価などがあげられる。

参 考 文 献

- 1) 株式会社三菱総合研究所：アジアオープンソースソフトウェア人材育成基盤整備事業報告書「OSS 技術者のスキルセット調査」(2005)。
- 2) 濱野賢一朗，鈴木友峰：オープンソースソフトウェアの本当の使い方，技評 SE 新書 (2007)。
- 3) 独立行政法人情報処理推進機構：2006 年度オープンソースソフトウェア活用基盤整備事業 OSS 技術教育のためのモデルカリキュラムに関する調査「調査 3：我が国の教育・

研修機関が提供する OSS 技術教育の現状とギャップ分析」(2007)。

- 4) Screencast. <http://en.wikipedia.org/wiki/Screencast>
- 5) vnc2swf Screen Recorder. <http://www.unixuser.org/~euske/vnc2swf/>
- 6) 浦尾 彰，三輪和久：作成プロセスの追従による組立てスキルの学習支援，信学論，Vol.J89-D, No.6, pp.1269-1278 (2006)。
- 7) 浦尾 彰，三輪和久：動画を用いたインタラクティブな学習環境における認知負荷の効果の検討，信学論，Vol.J91-D, No.2, pp.368-376 (2008)。
- 8) Sleipnir. <http://www.fenrir.co.jp/sleipnir/>
- 9) JAVA で HelloWorld (JSP 編) . <http://www.hellohiro.com/jsp.htm>
- 10) Crescent Eve. <http://www.kashim.com/eve/>
- 11) 田中 敏，山際勇一郎：ユーザーのための教育・心理統計と実験計画法，教育出版 (1992)。
- 12) XOOOPS のインストール (LINUX 編) . <http://kisa.homeip.net/wiki>
- 13) 黒須正明：ユーザビリティテストング，共立出版 (2003)。
- 14) Kurosu, H.M. and Sato, D.: A new data collection method for usability testing nem: Novice expert ratio method, *Usability Professionals' Association Conference 2002-Humanizing Design*, Posters (2002)。
- 15) 川田敏則，味岡義明：X ウィンドシステムを用いた教育支援システム，情報処理学会研究報告，Vol.94-CE, No.32, pp.9-16 (1994)。
- 16) 入部百合絵，藤原 真，安田孝美，横井茂樹：学習者の操作プロセスに適応した対話型ソフト学習システム，信学論，Vol.J91-D, No.2, pp.269-279 (2008)。
- 17) 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，Vol.J88-D-I, No.1, pp.45-55 (2005)。
- 18) 香山瑞恵，岡本敏雄：知識共有を目的としたドキュメント操作のための協調プラットフォーム，信学論，Vol.J91-D, No.2, pp.152-165 (2008)。
- 19) Screen Motion 3. <http://www.mswave.co.jp/>
- 20) SOFT SIMULATOR. <http://www.patolis.co.jp/products/system/softsimulator/>
- 21) ViewletBuilder5. <http://www.webdemo.co.jp/vb5/>
- 22) 海保博之 (編) : 認知心理学，朝倉書店 (2005)。
- 23) 海保博之：くたばれ，マニュアル！—書き手の錯覚，読み手の癩癩，新曜社 (2002)。

(平成 20 年 5 月 27 日受付)

(平成 20 年 12 月 5 日採録)



村上 幸一（学生会員）

平成 9 年防衛大学校情報工学科卒業。同年陸上自衛隊勤務。平成 15 年防衛大学校理工学研究科前期課程修了。同年高松工業高等専門学校電気情報工学科助手。現在、岡山大学大学院自然科学研究科後期課程在学中。教育工学，Web 技術に関する研究に従事。ACM，電子情報通信学会各会員。



船曳 信生（正会員）

昭和 59 年東京大学工学部計数工学科卒業。同年住友金属工業（株）勤務。平成 3 年ケースウエスタンリザーブ大大学院修士課程修了。平成 6 年大阪大学基礎工学部情報工学科講師。平成 7 年同助教授。平成 12 年カリフォルニア大学サンタバーバラ校客員研究員。平成 13 年岡山大学工学部通信ネットワーク工学科教授。平成 17 年同大学大学院自然科学研究科教授。博士（工学）。ネットワーク，最適化アルゴリズム，画像処理，教育工学，Web 技術，セキュリティ等に関する研究に従事。IEEE，電子情報通信学会各会員。



徳永 秀和（正会員）

昭和 61 年東京工業大学大学院総合理工学研究科修士課程修了。同年新日本製鐵（株）勤務。平成元年技術研究組合国際ファジィ工学研究所出向。平成 5 年高松工業高等専門学校制御情報工学科講師。平成 17 年高松工業高等専門学校制御情報工学科助教授。現在，高松工業高等専門学校制御情報工学科准教授。博士（工学）。機械学習，自然言語処理，Web 技術等に関する研究に従事。人工知能学会，日本知能情報ファジィ学会各会員。



重田 和弘（正会員）

昭和 63 年山口大学大学院工学研究科修士課程電子工学専攻修了。同年日本電信電話（株）勤務。平成 9 年東京工業大学大学院総合理工学研究科修士課程修了。平成 12 年東京工業大学大学院社会理工学研究科博士後期課程人間行動システム専攻修了。同年高松工業高等専門学校電気工学科講師。平成 14 年同校電気情報工学科助教授。平成 19 年同准教授。博士（工学）。教育工学に関する研究に従事。電子情報通信学会，日本教育工学会各会員。



中西 透（正会員）

平成 7 年大阪大学大学院基礎工研究科博士前期課程修了。平成 10 年同博士後期課程退学。同年岡山大学工学部情報工学科助手。平成 12 年同通信ネットワーク工学科助手。平成 15 年同講師。平成 17 年同大学大学院自然科学研究科講師。平成 18 年同助教授。平成 19 年同准教授。情報セキュリティ，ネットワークセキュリティに関する研究に従事。博士（工学）。電子情報通信学会会員。