

博士論文

機能実行履歴を用いたソフトウェア利用知識の共有と
その支援システム

森崎 修司

2001年2月13日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
博士(工学) 授与の要件として提出した博士論文である。

森崎 修司

審査委員： 井上 克郎 教授
 小山 正樹 教授
 関 浩之 教授
 松本 健一 助教授

機能実行履歴を用いたソフトウェア利用知識の共有と その支援システム*

森崎 修司

内容梗概

今日、幅広いユーザ層を想定したアプリケーションソフトウェアの多くは非常に多機能となっているが、実際に利用されている機能はそのごく一部に留まっている。その理由の一つは、利用知識を獲得する（ソフトウェアが提供する機能の存在に気づき、機能実行のための操作方法を理解し、その機能がどのような状況で有効に働くか理解する）コストが大きいことにある。ソフトウェアの多機能化に伴いマニュアルなども肥大化しており、特に、ユーザがその存在に気づいていない機能（未知機能）の利用知識を獲得することは困難である。

本論文では、ソフトウェアが提供する機能の利用知識の獲得コストを小さくすることを目的として、ユーザ間で機能実行履歴を共有する方式（CLAS: Community-based Learning for Application Software）を提案する。CLASでは似通った目的を持ち同一ソフトウェアを使用する複数のユーザから機能実行履歴を自動的に収集し、機能毎に各ユーザの実行頻度を付加した上でユーザ間で参照可能とする。機能毎の実行頻度を他ユーザと比較することにより、ユーザは未知機能を発見できる。

被験者全員に同一タスクを与え、機能実行履歴を共有する実験では、5名の被験者全員がのべ10個の機能の利用知識を獲得した。また、被験者の実作業環境で1～10ヵ月間収集した機能実行履歴を共有する実験では、6名の被験者全員がのべ60個の機能の利用知識を獲得した。実験で使用した共有支援システムは、対象ソフトウェアをMicrosoft Officeとする。大量の機能実行履歴から利用知識を効率よく獲得できるよう、他ユーザの履歴には含まれるが、ユーザ自身は未実行の機能を未知機能の候補として提示する。また、ユーザが指定した機能を含む部分系列を履歴から抽出することもできる。

* 奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 博士論文, NAIST-IS-DT9861021, 2001年2月13日.

キーワード

操作履歴，ユーザ支援，知識の共有，協調的情報フィルタリング，協調学習

Sharing Software Usage Knowledge and Its Support System by Using Histories of Function Executions *

Shuuji Morisaki

Abstract

Latest software applications for diverse users provide a very large number of functions. However, only limited functions are utilized. The cost of obtaining usage knowledge of software functions prevents users from learning more functions. The usage knowledge of software functions is to aware existence of a function, to learn operation to execute the function and to understand the usage context of the function. The growth of the number of functions involves the growth of the pages of operation manual. Users hardly find a function that is not expected by users.

In this thesis, CLAS (Community-based Learning for Application Software) is proposed. CLAS enables us to reduce the cost of obtaining usage knowledge of software functions by sharing history of function executions. CLAS automatically collects histories of function executions from users having similar purposes. Each function and its execution frequency contained in collected histories of function executions are available to users by CLAS. By comparing execution frequencies of users, a user can find functions which is unknown to the user.

An experiment shows that five subjects obtained usage knowledge of 11 functions. Another experiment shows six subjects obtained usage knowledge of 60 functions from histories of function executions in daily use of software. The system used in the experiments collects histories of function executions of Microsoft Office 2000. The system enables users to find usage knowledge of functions from a large number of histories by indicating users candidate functions of unknown functions. Candidate functions are

* Doctor's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT9861021, February 13, 2001.

not contained in a user's history but contained in other users histories. By specifying a function contained in histories of function executions, user can recognize subsequences which contain the specified function provided by the system.

Keywords:

operation history, user support, knowledge sharing, collaborative information filtering, collaborative learning

研究業績

1. 森崎修司, 門田暁人, 松本健一, 井上克郎, 鳥居宏次: 機能実行履歴を用いたソフトウェア利用知識の共有, 情報処理学会論文誌, vol.41, No.10, pp. 2770-2781. 2000.
2. 森崎修司, 白石裕美, 大和正武, 門田暁人, 松本健一, 鳥居宏次: 機能実行履歴を用いたソフトウェア機能の発見支援システム, 電子情報通信学会論文誌 (2001年6月号掲載予定).
3. Shuuji Morisaki, Kumiyo Nakakoji, Shingo Takada, Ken-ichi Matsumoto, Koji Torii: AID-Sim: Architecture for Integrating Distributed Software Development Simulators, Proceedings of International Symposium on Future Software Technology, pp. 137-142, 1998
4. Shuuji Morisaki, Akito Monden, Ken-ichi Matsumoto, Katsuro Inoue, and Koji Torii: Two-level user interface evolution by sharing usage logs, International Workshop on Principles of Software Evolution '99, pp.70-73, 1999.
5. 森崎 修司, 花川 典子, 松本 健一, 鳥居 宏次: 作業習熟を考慮したソフトウェア作業進捗モデル, 日本ソフトウェア科学会ソフトウェア工学の基礎ワークショップ, pp. 75-82 1997.
6. 中島田 義敬, 森崎 修司, 門田 暁人, 松本 健一, 鳥居 宏次: ソフトウェア開発行動の分析支援を目的とした映像検索システム, 電子情報通信学会技術研究報告, ソフトウェアサイエンス研究会, Vol. 96, No. 601, pp. 1-8 1997.
7. 森崎 修司, 中島田 義敬, 松本 健一, 鳥居 宏次: 行動履歴データとの同期再生機能をもつビデオ映像再生システムの試作, 電子情報通信学会全国大会講演論文集, 情報・システム, D-3-9, p. 67, 1997.
8. 伊藤充男, 森崎修司, 門田暁人, 松本健一, 鳥居宏次: デバッグ時間の短縮を目的とする二人によるデバッグングの実験的考察, 電子情報通信学会技術研究報告, 人工知能と知識処理研究会, Vol. 98, No. 634, AI98-82, pp.45-52, 1999.
9. 森崎修司, 伊藤充男, 門田暁人, 松本健一, 鳥居宏次: 二人によるソフトウェアデバッグにおける役割分担と情報交換, 情報処理学会研究報告, グループウェア研究会, Vol.99, No.40, 99-GW-32-7, pp.35-40, 1999.

10. 白石裕美, 森崎修司, 門田暁人, 松本健一, 井上克郎: 未知機能の発見支援を目的としたソフトウェア機能実行履歴の提示, 情報処理学会研究報告, グループウェア研究会, Vol. 2000, No. 97, 2000-GW-37-3, pp.13-18, 2000.

目次

1. 序論	1
2. CLAS	4
2.1 概要	4
2.2 CLASが共有を支援するソフトウェア機能	5
2.3 利用知識共有の手順	7
2.4 機能実行履歴の収集方式	9
2.5 機能実行履歴の提示	12
2.6 コミュニティ内のユーザ選択	14
3. CLAS方式の評価実験	16
3.1 概要	16
3.2 タスク	16
3.3 試作システム	19
3.4 実験手順	21
3.5 結果	22
3.6 考察	26
4. 利用知識共有支援システム	30
4.1 システムの要件	30
4.2 構成	31
4.3 機能実行履歴の収集	34
4.4 機能実行履歴の提示	35
4.4.1 提示画面	35
4.4.2 データの流れ	43
5. 評価実験	47
5.1 評価実験(同一タスク)	47
5.1.1 概要	47
5.1.2 タスク	47
5.1.3 実験手順	49

5.1.4	結果	50
5.1.5	評価実験(同一タスク)の考察	52
5.2	評価実験(類似タスク)	53
5.2.1	概要	53
5.2.2	タスク	54
5.2.3	結果	55
5.2.4	評価実験(類似タスク)の考察	57
6.	関連研究	59
6.1	知識共有	59
6.2	操作履歴	59
6.3	ユーザ支援	60
7.	結論	62
	参考文献	65
	謝辞	70
	付録	73

目 次

1	CLAS の概要	5
2	ソフトウェアが提供する機能の集合とユーザが既知の機能の集合	6
3	ソフトウェアが提供する機能の集合と有用な機能の集合	6
4	ユーザ $U(i)$ の未知機能の分類	6
5	収集方式 (1): プラグイン方式	10
6	収集方式 (2): 対象ソフトウェア変更方式	10
7	収集方式 (3): GUI ツールキット変更方式	11
8	ユーザに提示する機能の集合 (参照するユーザの実行頻度が小さい機能)	13
9	ユーザに提示する機能の集合 (参照するユーザ以外の実行頻度が小さい機能)	14
10	ユーザに提示する機能 (特定のユーザの実行頻度が大きい機能)	15
11	被験者に与えられるファイルの内容	17
12	被験者に与えられるタスク指示書	18
13	試作システムの構成	20
14	試作システムの機能実行履歴提示部の画面例	21
15	各被験者の利用知識の獲得時期	24
16	システムの構成	33
17	機能実行履歴収集のデータの流れ	33
18	機能実行履歴の例	34
19	機能毎の実行頻度の提示方法	36
20	機能毎の実行頻度の提示方法	37
21	機能毎の実行頻度一覧の画面例	38
22	並べ替えの例 (第 1 キー: 他ユーザによる実行率 (降順), 第 2 キー: 履歴参照ユーザの実行頻度 (昇順))	39
23	並べ替えの例 (第 1 キー: 他ユーザによる実行率 (昇順), 第 2 キー: 履歴参照ユーザの実行頻度 (昇順))	40
24	並べ替えの例 (第 1 キー: 実行頻度の合計 (降順), 第 2 キー: 履歴参照ユーザの実行頻度 (昇順))	41
25	部分系列の提示方法	41
26	部分系列に含まれる同一機能の色分け	42

27	指定機能を中心とした部分系列一覧の画面例	43
28	蓄積した機能実行履歴の属性一覧の画面例	44
29	機能実行履歴提示のデータの流れ	45
30	機能実行履歴の属性のフォーマット	46
31	各被験者の利用知識の獲得時期	51

表 目 次

1	被験者の Word 使用経験	23
2	被験者の作業時間, 実行機能数, 機能実行回数	25
3	同一タスクによる評価実験の被験者の Word 使用経験	48
4	各機能により省ける操作の概数	49
5	類似タスクによる評価実験の被験者の Word 使用経験	53
6	類似タスクによる評価実験の被験者の PowerPoint 使用経験	54
7	実験に使用した Word 2000 の機能実行履歴	55
8	実験に使用した PowerPoint 2000 の機能実行履歴	55
9	実験結果 (Word)	56
10	実験結果 (PowerPoint)	57
11	機能実行履歴と抽出する機能実行履歴の部分系列の例	74

1. 序論

幅広いユーザ層，ならびに，使用目的を想定したアプリケーションソフトウェアが提供する機能の数は増加の一途をたどっている．例えば，Microsoft Word 2000 の場合，メニュー項目数，及び，コンテキストメニュー¹の項目数を合計するとその数は 500 以上となる．

しかし，ソフトウェアの提供する全ての機能がユーザによって利用されているわけではない．我々が行った予備実験において，A4 判で約 190 ページ（文字数約 10 万，図表数 72）のソフトウェア設計書を Microsoft Word 2000 で作成する作業において，利用された機能数（メニュー項目数と GUI ボタン数の合計）は約 70，即ち，全体の約 7% であった [25][47]．また，文献 [44] では，UNIX のシェルの一つである tcsh が提供する約 50 個のコマンド編集機能のうち，実際にユーザが使う機能（利用機能）は全体の 50% にも満たないことが報告されている．

一般に，利用機能数はソフトウェアの使用時間と共に増加するが，ある程度まで増加すると頭打ちになることが知られている．例えば，文献 [46] の実験では，UNIX のシェルインタプリタである tcsh が提供する機能数の約 30% 程度までしか利用機能数が増加しないことが確かめられている．しかも，利用すれば作業効率がより高くなる機能がソフトウェアによって提供されているにも関わらず，それら機能の存在をユーザが積極的に知ろうとするとは限らないことも観察されている [4]．

利用機能数が頭打ちとなる原因の一つは，ソフトウェアが提供する膨大な機能から，自分にとって有用な（作業効率を高める）機能の利用知識を獲得するコストが高いことにある．ここで，利用知識とは，実際に機能を実行し試行錯誤したり，オンラインヘルプなどのマニュアルを参照したりしながら，機能を発見し，機能を実行するための操作方法を理解し，その機能が有効に働くコンテキストを理解することである．いったんユーザが機能の利用知識を獲得すれば，その機能が適切に働く状況になったときは常に利用可能となり，ユーザの作業効率は高くなる．ただし，利用可能である機能数が増加するにつれて，機能の発見による作業効率の上昇幅は徐々に小さくなる．機能の利用知識獲得コストが一定であるとする，作業効率の上昇幅がある値以下になれば，新たな機能の利用知識を獲得する必要性をユーザは感じにくくなる．新たな機能を利用しなければ必要な作業が遂行できない場合は別であるが，既知の機能やその組み合わせでユーザは作業を遂行するようになる [35]．

¹ 状況依存で現れるメニュー．MS Windows では，マウスの右ボタンをクリックすると現れる

利用機能数が頭打ちとなるもう一つの原因は、ソフトウェアが提供する機能、あるいは、提供するかもしれない機能の存在自体をユーザが想像できるとは限らないことにある。特に、ソフトウェアの利用経験の少ないユーザには、存在自体を想像できない機能が多くある。例えば、文書作成ソフトウェアにおいて、利用経験の少ないユーザが、文書中のフォントだけを置換する機能を想像し、ソフトウェアが提供していると期待するとは考えにくい。最近では、ソフトウェアの多くがオンラインヘルプシステムを持ち、ヘルプに対する質問文検索やキーワード検索を可能としている。しかし、存在自体が想像できない機能が検索結果となるようなキーワードを選定することは容易ではない [3]。また、ソフトウェア本体の多機能化に伴って大規模化しているマニュアルの中から、存在すら想像できない機能の利用知識を獲得することは多くのユーザにとって容易なことではない。

ユーザにソフトウェア機能の存在を積極的に知らせ、機能の利用知識獲得コストを小さくすることを目的としたユーザ支援方式の一つに active help system [10][32][48][49][50] がある。Active help system ではユーザの操作履歴に基づいてユーザの意図を推測するためのユーザモデルを持ち、ユーザが気づいていない(と推測される)機能を知らせる。Activist[10]、Ehelp[32]、Anchises[50] では、テキストエディタの提供する編集、カーソル移動機能の利用知識の獲得を積極的に支援することを目的としている。これらのシステムは、同一の結果をもたらす機能の組合せ、及び、その操作数をユーザモデルとして持つ。ユーザがある機能を実行する度に、より少ない操作数で同じ結果を得られる機能の存在を調べ、存在する場合にはその機能の存在をユーザに知らせる。これらのシステムでは、支援対象としている機能数が高々50程度であるため、利用知識獲得の効果的な支援が可能であるが、本論文で対象とする機能が肥大化したソフトウェアでは、効果的なユーザモデルの構築、及び、適切な支援を行うことは容易ではない [9]。

本論文では、ユーザモデルを構築することなくソフトウェア機能の利用知識獲得コストを小さくすることを目的として、似通った使用目的で同一ソフトウェアを使用するユーザ間で利用知識を共有するための方式 *CLAS* (Community-based Learning for Application Software) を提案する。ただし、機能に関する知識は、いわゆる「暗黙知」の一種である。共有可能な形式とすることはもちろん、ユーザから知識の提供を受けること自体が容易ではない [11]。そこで、各ユーザの実行したソフトウェア機能を機能実行履歴(コマンド実行履歴)として有形化した上で収集し、それをユーザ間で相互参照可能とする。また、個々のユーザにとって有用な機能を選択してユーザに提供する、あるいは、ユーザ

が容易に選択できるしくみとして機能実行履歴の提示方法を提案する．具体的には，対象ソフトウェアの特性に応じた3つの機能実行履歴の収集方式，及び，あるユーザが他のユーザの機能実行履歴から未知の機能を発見することを支援する提示方法を提案する．

以降，2章で機能の利用知識を共有する方式 *CLAS* について述べ，3章においてその評価実験について述べる．4章では，3章の評価実験の結果や被験者から得られたインタビューを踏まえて開発したシステムを紹介する．5章でシステムを用いた二つの評価実験について述べる．6章で関連研究を紹介し，7章でまとめる．

2. CLAS

本章では，CLASの概要について述べた後，CLASが利用知識の共有を支援できるソフトウェア機能の分類について述べる．次に，利用知識を機能実行履歴として有形化する方法を述べ，その収集方式，及び，提示方式について述べる．

2.1 概要

CLASでは似通った目的で同一ソフトウェアを使用する複数のユーザでコミュニティを形成する．そのコミュニティの間で既知の機能を相互に参照することにより，あるユーザは未知の機能を同一コミュニティに属する他ユーザの既知の機能から発見する．

CLASでは，コミュニティ内のあるユーザが実行したことの機能はそのユーザにとって既知の機能であると仮定し，コミュニティ内のユーザ間で実行したことの機能を相互参照する．あるユーザは，コミュニティ内の他のユーザが実行したことの機能から自分にとって未知の機能を発見できる．同一コミュニティに属するユーザ間では有用な機能も似通っていると考えられるため，ソフトウェアが提供する膨大な機能の全てを探すよりも少ないコストで有用な機能を発見することができる．

ユーザが実行した機能はソフトウェア機能実行履歴として有形化する．機能実行履歴とは，ソフトウェア利用時にユーザが実行した機能名，具体的には選択されたメニュー項目やGUIボタンの名称を，ユーザ毎，ソフトウェア毎にその実行日時と共に時系列に並べたものである．

CLASの概略を図2.1に示す．機能実行履歴を相互参照しようとするユーザは「機能実行履歴収集サーバ」とネットワークで結ばれた計算機上でソフトウェアを利用する．すると，ユーザの計算機上で稼動する「機能実行履歴収集部」が機能実行履歴を収集し，ネットワークを介して「機能実行履歴収集サーバ」へと送出する．機能実行履歴の収集と送出は自動的，かつ，ユーザの作業を妨げないように行われる．「機能実行履歴収集サーバ」はユーザからの要求に応じて，蓄積した機能実行履歴をネットワーク経由でユーザに提供する．ユーザの計算機上で稼動する「機能実行履歴提示部」は，提供された機能実行履歴における各機能の実行頻度の提示，ユーザが指定した機能を含む機能実行履歴の部分系列の提示を行い，ユーザ間で使用機能の相互参照を支援する．

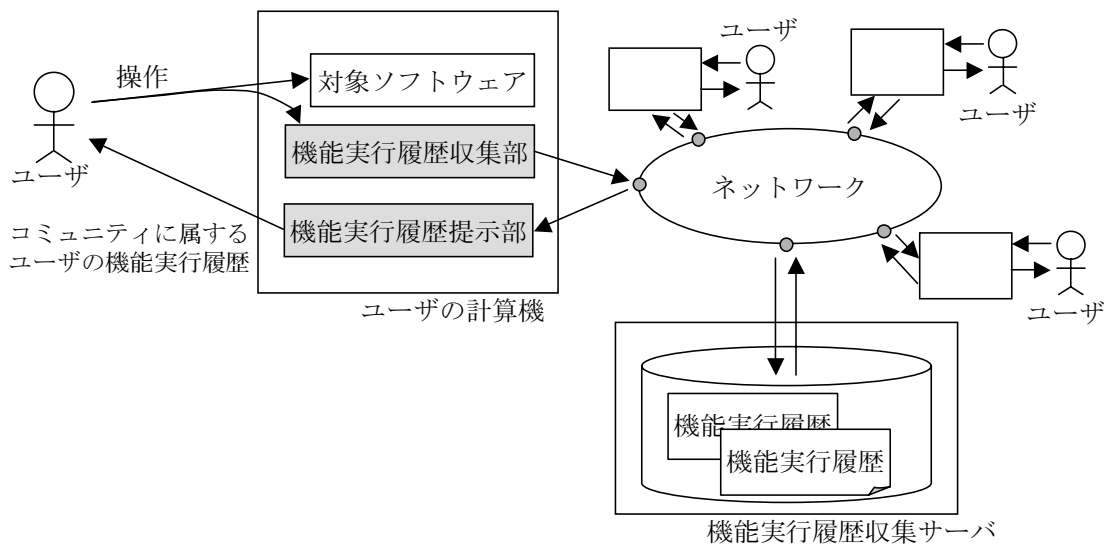


図 1 CLAS の概要

2.2 CLAS が共有を支援するソフトウェア機能

図 2 はソフトウェアが提供する機能 F_S の集合とユーザー $U(i)$ にとって既知の機能の集合 $F_{U(i)}$ との関係を表している。CLAS が対象とする機能が肥大化したソフトウェアでは、通常、図 2 のようにソフトウェアが提供する機能の集合よりもユーザーにとって既知の機能の集合のほうが小さくなることが多い。CLAS の目的は $F_{U(i)}$ に属する有用な（作業効率を高める）機能の数を大きくすることである。

図 3 はあるユーザー $U(i)$ が利用知識を持つ機能 ($F_{U(i)}$) とコミュニティに属するいずれかのユーザーが利用知識を持つ機能 (F_C) との関係を表している。コミュニティは似通った目的で同一のソフトウェアを使用するユーザーから構成される。 F_C はコミュニティに属するユーザーのいずれかが利用知識を獲得し、実行する機能の集合である。すなわち、 F_C は $F_{U(i)} (i = 1, \dots, n. n: \text{コミュニティ } C \text{ に属するユーザーの数})$ の和集合である。差集合 $F_C - F_{U(i)}$ に含まれる機能はあるユーザー $U(i)$ にとって有用な機能である可能性が高い。その理由は、似通った目的を持ち同一のソフトウェアを使用するユーザーが属するコミュニティ内では、各ユーザーにとって有用な機能は似通っていると考えられるからである。

図 4 は CLAS がユーザーに提示すべき機能を、文献 [9] で定義された 3 つのタイプの機能を用いて分類している。通常、タイプ 1 やタイプ 2 に属する機能はオンラインヘルプを

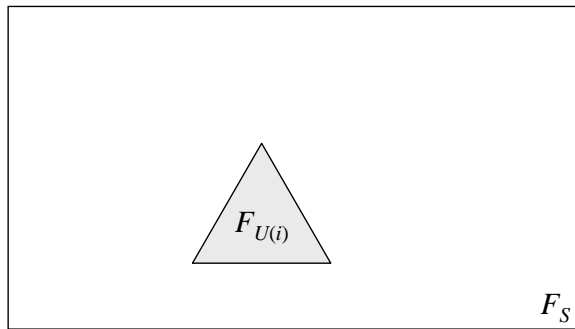


図 2 ソフトウェアが提供する機能の集合とユーザが既知の機能の集合

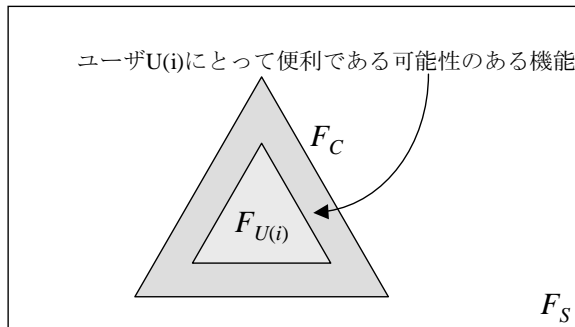


図 3 ソフトウェアが提供する機能の集合と有用な機能の集合

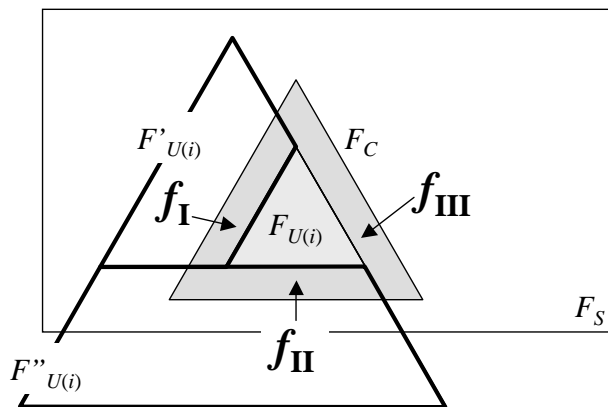


図 4 ユーザ $U(i)$ の未知機能の分類

はじめとするマニュアルから見つけ出すことが可能である。CLASでは、タイプ1、タイプ2に加えて、マニュアルから探し出すことが難しいタイプ3の機能の利用知識の獲得を支援することができる。

タイプ1 $f_I = F'_{U(i)} \cap (F_C - F_{U(i)})$ ソフトウェア S が提供し、ユーザ $U(i)$ が存在は知っているが、滅多に使わない機能である。ユーザ $U(i)$ はこの機能の使用手順やどのようなコンテキストで使用すると効果的であるかを知らない。ユーザ $U(i)$ は機能の具体的な名前や概要を知っているので、オンラインヘルプなどのマニュアルを利用することによりこのタイプの機能に関する説明などを得ることができる。

タイプ2 $f_{II} = F''_{U(i)} \cap (F_C - F_{U(i)})$ ソフトウェア S が提供し、ユーザ $U(i)$ はそのことを期待しているが、実際には1度も実行されたことがない機能である。ユーザ $U(i)$ が機能の名前やその機能の特徴づけるキーワードを思い浮かべることができれば、オンラインヘルプなどのマニュアルを利用することによりこのタイプの機能に関する説明を得ることができる。

タイプ3 $f_{III} = (F'_{U(i)} \cup F''_{U(i)})^C \cap (F_C - F_{U(i)})$ ソフトウェア S が提供するが、ユーザ $U(i)$ はその存在に気づいていない機能である。ユーザ $U(i)$ は気づいていない機能に対して、その機能を探することができる適切なキーワードや質問文を思いつくことは容易ではないので、オンラインヘルプの検索機能を使用しても、このタイプの機能をマニュアルから探すことは難しい。

ここで、 $F'_{U(i)}$ はコミュニティ C のユーザ $U(i)$ がその存在を漠然と知っており、必要があれば実行できる機能である。ただし、実行するにはオンラインヘルプなどのマニュアルを使用し、機能の説明を得る必要がある機能の集合である ($F_{U(i)} \cap F'_{U(i)} = \phi$, $F'_{U(i)} \subseteq F_S$)。 $F''_{U(i)}$ はユーザ $U(i)$ がアプリケーションが提供すると期待する機能の集合である ($F_{U(i)} \cap F''_{U(i)} = \phi$, $F'_{U(i)} \cap F''_{U(i)} = \phi$)。

2.3 利用知識共有の手順

CLASの利用知識共有の手順は次の3つの手順からなる。

手順1 ソフトウェアの機能実行履歴の収集

コミュニティ内でユーザの機能実行履歴を収集する。機能実行履歴には次の3つが含まれる。

- 実行された機能の識別子
- 機能実行したユーザの識別子
- 機能を実行した日時

ユーザが普段どおりソフトウェアを実行することにより、ユーザの計算機上で稼働する機能実行履歴収集部が機能実行履歴を収集する。機能実行履歴収集部は、ユーザの操作を監視し、そのうち、機能実行に関するものだけを抽出する。収集した機能実行履歴を機能実行履歴サーバに蓄積しておき、次のステップで同一コミュニティ内の他のユーザの機能実行履歴と比較する。

手順 2 未知機能の提示

コミュニティ内のユーザは機能実行履歴サーバに要求することにより、いつでも、同一コミュニティ内の他のユーザの機能実行履歴を取得することができる。

同一コミュニティ内のユーザは自らの機能実行履歴と他ユーザの機能実行履歴とを比較することにより、あるユーザにとって未知である機能を他のユーザの機能実行履歴から発見することができる。例えば、あるユーザが実行したことの無い機能をコミュニティ内の多くのユーザが実行したことがあれば、有用な機能であるが、そのユーザはその存在に気づいていない可能性がある。これらの機能は前述のタイプ 2、及び、タイプ 3 に属する機能であり、オンラインヘルプをはじめとするドキュメントから探し出すことが難しい機能である。

手順 3 機能に関する情報の取得

ユーザは同一コミュニティ内の他のユーザの機能実行履歴から発見した機能の詳しい情報を、必要であれば、以下の方法により得ることができる。

- 実際に機能を実行してみる。
実際に機能を実行することにより、その機能のはたらきを理解することができる。
- 発見した機能名を手がかりにマニュアルを参照する。
機能名をキーワードとして、オンラインヘルプを検索したり、マニュアルの索引を使用したりして機能のはたらきを理解することができる。
- 同一コミュニティ内の他のユーザがその機能の前後にどのような機能を実行しているかみる。

機能実行履歴の部分系列のうち、発見した機能を含む部分系列をみることにより、その機能を実行するコンテキストを理解できる。

手順 1 においてユーザが行うタスクは同一タスク、類似タスクの 2 種類があり、機能実行履歴の蓄積量、利用知識の共有の目的に応じて使い分ける。

- 同一タスク

機能実行履歴の蓄積が小さい場合に行なう。コミュニティ内の各ユーザが予め定められた同一のタスクを遂行し、タスク遂行時の機能実行履歴を相互参照する。相互参照する履歴は、同一タスクを遂行した際に実行されたものに限定する。コミュニティで共通する目的を持つユーザにとって有用な機能をなるべく多く含むようにタスクを設定することが望ましい。

同一タスクを行った際の機能実行履歴の相互参照は、同一組織内など普段から似た作業を行っているコミュニティで知識を共有したい場合に特に有効である。予め定めるタスクは、普段行う作業を数多く含むように設定することが望ましい。

- 類似タスク

機能実行履歴の蓄積が一定以上ある場合に行なう。コミュニティ内のユーザの普段の機能実行履歴を蓄積し、それを相互参照する。タスクは特に設けないが、2.1 節で述べたように似通った目的をもつユーザでコミュニティを形成するので、似通ったタスクを遂行した際の実行履歴を相互参照することになる。特定のタスクをユーザがわざわざ行う必要がない上、機能実行履歴収集のためのコストが極めて低い。

2.4 機能実行履歴の収集方式

ソフトウェアの機能実行履歴を収集する三つの方式を図 5, 6, 7 に示す。各方式では、グラフィカルユーザインタフェース (GUI) を持つソフトウェアを前提とし、GUI ツールキットにより呼び出されたコード (メソッド) の履歴を機能実行履歴として収集できる。一般に、GUI に対するユーザの操作 (キー入力、マウス操作など) の情報は、まず OS が受け取り、GUI ツールキットと呼ばれるミドルウェアを介してアプリケーションソフトウェアに伝えられる。GUI ツールキットは、ユーザの操作と実行されるメソッドとの対応を管理し、ユーザがある GUI 操作を行った時に、その操作に対応したメソッドを呼び出す機能を持つ。GUI ツールキットの例としては、Microsoft Windows におけ

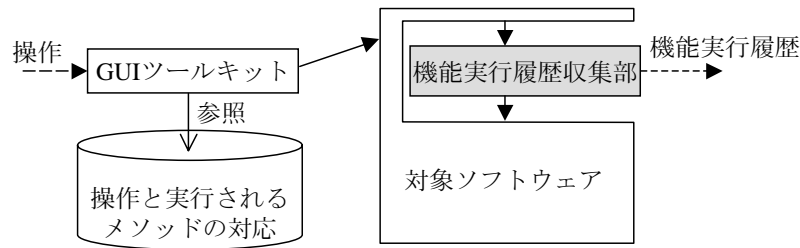


図 5 収集方式 (1): プラグイン方式

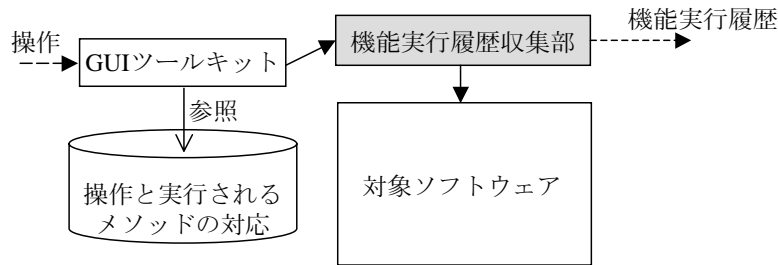


図 6 収集方式 (2): 対象ソフトウェア変更方式

る MFC(Microsoft Foundation Class) , UNIX における XLib , XToolkit , GTK+(GNU Tool Kit) , Java における AWT (Abstract Window Toolkit) , Swing などが挙げられる . 本論文では , GUI ツールキットにより呼び出された個々のメソッドがソフトウェアの各機能に対応するとみなし , メソッドの実行履歴を機能実行履歴として収集する . 提案する三つの方式はそれぞれ適用範囲が異なるため , 対象ソフトウェアに応じいずれかの方式を選択する .

1. プラグイン方式

実行履歴収集部を , 対象ソフトウェアのプラグインモジュールとして実装し , ユーザが機能実行する度に呼び出されるようにする (図 5) .

対象ソフトウェアがプラグインモジュールを提供し , かつ , プラグインインタフェースからユーザの機能実行が監視できる場合に適用できる . 適用できるアプリケーションの例としては , Microsoft Office (Word , Excel , PowerPoint 等) , Adobe Photoshop , Illustrator などが挙げられる .

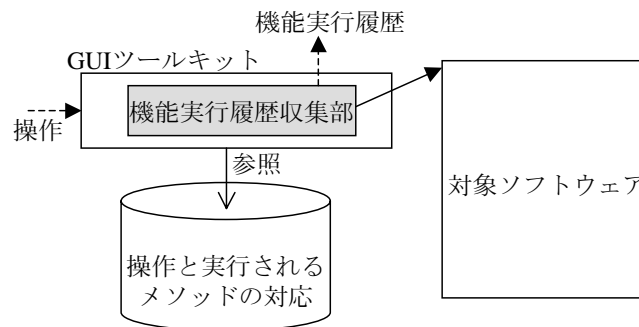


図 7 収集方式 (3): GUI ツールキット変更方式

本方式の利点は、あらかじめ定められたインタフェースに従ってプラグインモジュールを作成すればよいので実装が容易である点と、ソースコードが公開されていないアプリケーションソフトウェアに適用できる点である。欠点は、収集可能な機能実行履歴がプラグインインタフェースの仕様に依存するため、全ての実行機能を監視できるとは限らない点である。

2. 対象ソフトウェア変更方式

実行履歴収集部を、対象ソフトウェアに組み込み、機能実行のたびに呼び出されるようにする(図 6)。機能実行履歴収集部と対象ソフトウェアを開発する組織や個人が同一である場合や、ソースコードが公開されているソフトウェア(GPL: GNU General Public License に準拠したフリーソフトウェアなど)、ソースコードが入手できる場合に適用できる。

本方式の利点は、ソフトウェアに含まれる全ての機能の実行履歴を記録するように実行履歴収集部を実装可能な点である。欠点は、ソフトウェアのソースコードが公開されていない場合には適用できない点である。

3. GUI ツールキット変更方式

GUI ツールキットを変更することにより、実行履歴収集部を GUI ツールキットに組込む。具体的には、GUI ツールキットに実行履歴収集部を追加し、既存の GUI ツールキットと置き換える(図 7)。現存するほとんどの GUI ツールキットはソースコードが公開されており、ダイナミックリンクライブラリとして実装されている

ため、対象ソフトウェアのプラグインインタフェース、ソースコードの公開に関わらず適用できる。

本方式の利点は、ソフトウェア毎に機能実行履歴収集部を作成する必要がない点である。ある GUI ツールキットを変更して機能実行履歴収集部を組込むと、その GUI ツールキットを用いる全てのソフトウェアから機能実行履歴を収集できる。欠点は、GUI ツールキットが受取る GUI イベントの管理が複雑になり、実装にコストがかかる点である。一般に、GUI ツールキットは OS からユーザの全ての操作を受取り、ソフトウェア毎に配送する。そのため、GUI イベントを配送するためには実行中のソフトウェアを管理する必要がある。機能実行履歴収集部を GUI ツールキットに組込む本方式では、他の収集方式と異なり、管理部分も考慮にいたれた実装が必要となる。

2.5 機能実行履歴の提示

本節では、ユーザが機能実行履歴を相互参照する際に *CLAS* が提示すべき機能を述べる。機能実行履歴を参照するだけでも、ユーザは未知の機能を発見することができるが、本節で述べるような機能を提示することにより、有用であるがユーザがその存在に気づいていない機能の発見を支援することができる。

ユーザに提示すべき機能は次のとおりである。

- 他の多くのユーザが実行したことがあり、かつ履歴を参照するユーザの実行頻度が小さい機能

履歴を参照するユーザが実行したことの無い機能で、他の多くのユーザが実行したことがある機能は、履歴を参照するユーザが気づいていない有用な機能である可能性がある。図 8 はユーザ $U(1)$, $U(2)$, $U(3)$, $U(4)$ が実行したことがある機能の集合 $(F_{U(1)}, F_{U(2)}, F_{U(3)}, F_{U(4)})$ を図示したものである。履歴を参照するユーザを $U(1)$ とすると、 $\overline{F_{U(1)}} \cap (F_{U(2)} \cap F_{U(3)} \cap F_{U(4)})$ (図 8(a)) に含まれる機能をユーザ $U(1)$ が見ることにより、有用な機能の利用知識獲得を支援することができる。また、 $\overline{F_{U(1)}} \cap (\overline{F_{U(2)}} \cap F_{U(3)} \cap F_{U(4)})$ (図 8(b)) や $\overline{F_{U(1)}} \cap (F_{U(2)} \cap \overline{F_{U(3)}} \cap F_{U(4)})$ (図 8(c)) に含まれる機能もユーザ $U(1)$ にとって有用な未知機能である可能性がある。

- 履歴を参照するユーザしか実行したことの無い機能

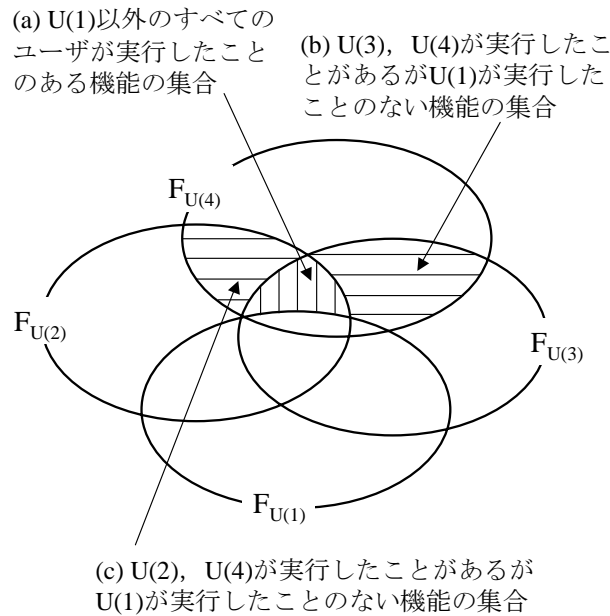


図 8 ユーザに提示する機能の集合 (参照するユーザの実行頻度が小さい機能)

履歴を参照するユーザしか実行したことのない機能を提示することにより、その機能の代替となる機能をそのユーザが発見する動機を与えることができる。 $F_{U(1)} \cap \overline{(F_{U(2)} \cup F_{U(3)} \cup F_{U(4)})}$ (図 9 中 (a)) には、ユーザ $U(1)$ だけが実行したことがある有用な機能、及び、ユーザ $U(1)$ にとって未知の有用な機能を実行したときと同様な結果を得られる機能が含まれる。ユーザ $U(1)$ が不便を感じている機能がこの集合に含まれていれば、その代替となる有用な機能が存在することを示唆することになる。

このような機能を提示することは、ユーザ間で機能の実行頻度が同程度となる同一タスク方式で履歴を参照する場合に特に有効である。

- 他ユーザによる実行頻度が高く、かつ、履歴を参照するユーザが実行したことのない機能

履歴を参照するユーザ以外のあるユーザだけが大きく実行している機能は、有用な機能であるが、他の多くのユーザにとって未知の機能である場合がある。図 10 は機能毎の実行頻度を棒グラフにしたものである。この例では、機能 B, C, D はそれ

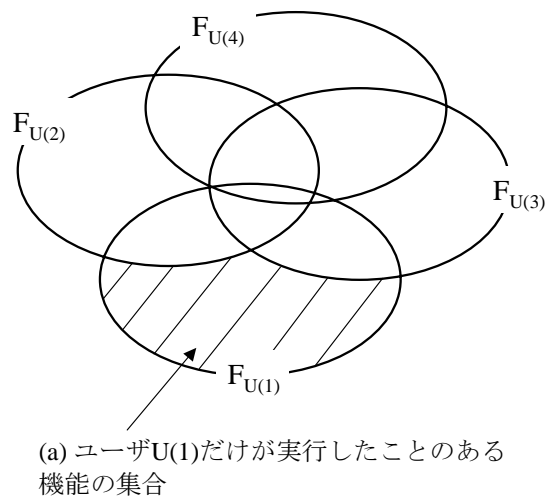


図 9 ユーザに提示する機能の集合 (参照するユーザ以外の実行頻度が小さい機能)

それ、ユーザ $U(4)$ 、 $U(2)$ 、 $U(3)$ による実行頻度が高く、かつ、それ以外のユーザの実行頻度が低い機能である。このような機能には、そのユーザだけが既知である有用な機能が含まれる場合がある。

2.6 コミュニティ内のユーザ選択

同一コミュニティ内のユーザのうち、特に参考にしたいユーザがいる場合や、履歴収集を始めたばかりのユーザなど履歴を参照するユーザから除外したいユーザがいる場合がある。コミュニティ内のユーザを選択し、選択したユーザの履歴を参照することにより、未知の有用な機能を発見することができる。

ユーザを選択する際の参考となるように、次の項目を各ユーザの機能実行履歴の属性として提示する。

- 使用機能の種類
- 履歴を参照するユーザと共通する機能数
- 機能実行履歴の行数
- 機能実行履歴の送信回数

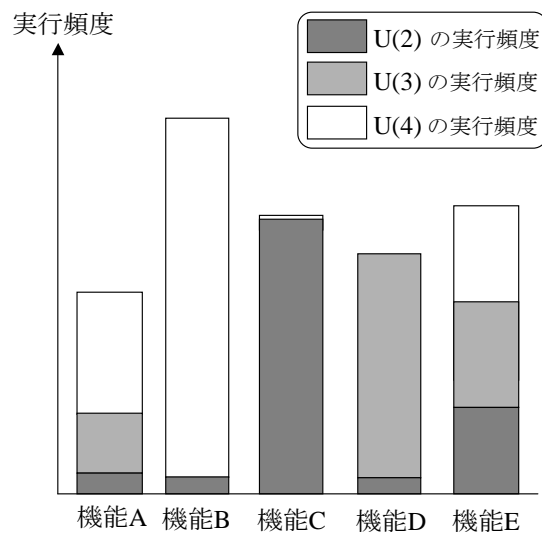


図 10 ユーザに提示する機能 (特定のユーザの実行頻度が大きい機能)

- 機能実行履歴の初回送信日時

これらの項目を提示することにより、他ユーザの履歴の属性から他ユーザの履歴が、履歴を参照するユーザにとって、どの程度参考になるかを知ることができる。例えば、履歴を参照するユーザの履歴とあるユーザの履歴とに共通して含まれる機能が多い場合には、そのユーザの履歴が特に参考になると考えられる。参考になるユーザの使用機能のうち、履歴を参照するユーザが実行していない機能があれば、それらの機能が履歴を参照するユーザにとって有用である機能の可能性が高い。

3. CLAS 方式の評価実験

CLAS 方式の妥当性を確認するために、評価実験を実施した。ただし、CLAS 方式の全ての妥当性を評価するのではなく、基礎となる「あるユーザが同一コミュニティ内の他ユーザの機能実行履歴から未知機能を発見でき、それがそのユーザにとって有用な機能であるか」という点に限定している。

3.1 概要

実験のあらまはは次のとおりである。

- 被験者は奈良先端科学技術大学院大学の教官 2 名，修士課程学生 3 名の計 5 名である。
- 被験者に与えられたタスクは、プレーンテキストをある形式の文書に整形することである。タスクの詳細は以降の節で述べるが、タスク中には 4 つのサブタスクが暗に（被験者には事前に知らせずに）設定されており、それらタスクを効率よく実行できる 4 つの機能の利用知識が、提案方式により獲得されるかどうかを評価する。
- 被験者は Microsoft Word 2000 を用いてタスクを実行する。Microsoft Word 2000 は現在広く利用されているワープロソフトの一つである。また、1 で述べたとおり、多機能ではあるが、ユーザが実際に利用している機能はその一部に過ぎない典型的なアプリケーションソフトウェアである。なお、各被験者の Microsoft Word の使用年数、主な作成ドキュメントを表 1 に示す。
- 機能実行履歴の収集、機能実行履歴の被験者への提示は、次節で述べる試作システムを用いて行った。
- 実験終了時に、被験者にアンケートとインタビューを行い、4 つのサブタスクを効率よく遂行できる 4 つの機能について、既知であったか、実験中に利用知識を獲得したか、サブタスク遂行のため利用したか、を確認した。

3.2 タスク

被験者に与えるタスクは共通で、Microsoft Word 2000 を用いて（ファイルとして格納されている）プレーンテキストをある形式の文書に整形する作業である。図 11 に被験

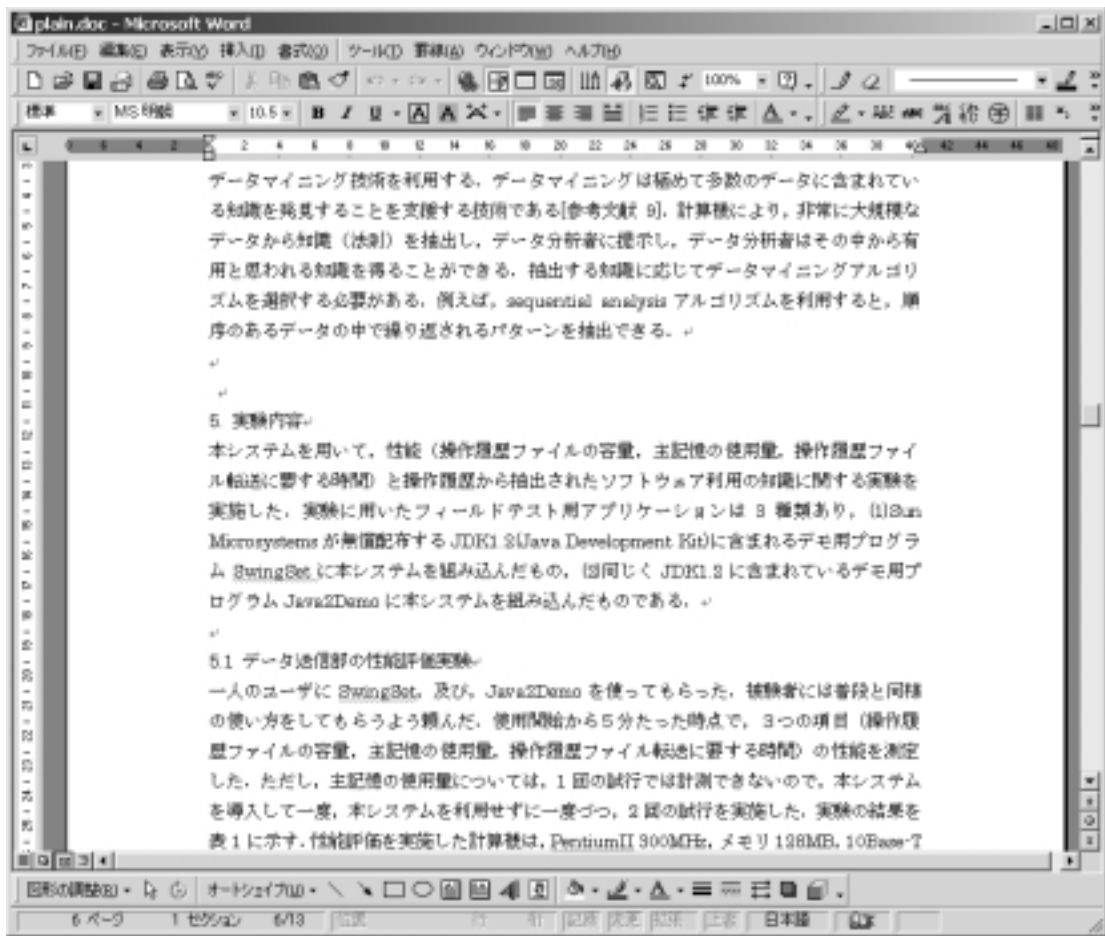


図 11 被験者に与えられるファイルの内容

者に与えたプレーンテキストを Microsoft Word 2000 で開いたときのスクリーンショットを示す。被験者には、プレーンテキストのファイル、整形済み文書（出来上がり見本）のハードコピーが渡される。整形すべき部分やその具体的内容は、出来上がり見本のハードコピーに指示として書き込まれている（図 12）。文書はあるシステムの解説書の一部であり日本語で書かれている。出来上がり時のサイズは A4 判 13 ページ、約 13,000 文字からなる。図は含まれていない。

整形作業は次の 4 つの作業（サブタスク）T1 ~ T4 から構成されている。

T1：目次の作成...章，及び，節見出しとページ番号の一覧を作成する。

5. 実験内容] MSゴシック714pt

本システムを用いて、性能(操作履歴ファイルの容量、主記憶の使用量、操作履歴ファイル転送に要する時間)と操作履歴から抽出されたソフトウェア利用の知識に関する実験を実施した。実験に用いたフィールドテスト用アプリケーションは3種類あり、(1)Sun Microsystemsが無償配布するJDK1.2(Java Development Kit)に含まれるデモ用プログラムSwingSetに本システムを組み込んだもの、(2)同じくJDK1.2に含まれているデモ用プログラムJava2Demoに本システムを組み込んだものである。

← イントロ(1ページ)

MSP明朝10.5pt

5.1 データ送信部の性能評価実験] MSゴシック11pt

一人のユーザにSwingSet、及び、Java2Demoを使ってもらった。被験者には普段と同様の使い方をしてもらうよう頼んだ。使用開始から5分たった時点で、3つの項目(操作履歴ファイルの容量、主記憶の使用量、操作履歴ファイル転送に要する時間)の性能を測定した。ただし、主記憶の使用量については、1回の試行では計測できないので、本システムを導入して一度、本システムを利用せずに一度ずつ、2回の試行を実施した。実験の結果を表1に示す。性能評価を実施した計算機は PentiumII 300MHz、メモリ198MB、10Base-Tのネットワークインターフェース

MSP明朝10.5pt

図 12 被験者に与えられるタスク指示書

T2 : ページ番号の付加...全てのページのフッタ中央にページ番号を付加する .

T3 : 表の作成 (3カ所) ...タブで区切られた文字列を指示されたとおり罫線で囲み表を作成する .

T4 : 章、及び、節見出しのフォント変更 (21カ所) ...文書全体を通じて、章と節の見出し部分のフォントを「MSゴシック」に変更する (本文のフォントは、MS明朝) .

Microsoft Word 2000 は、これら 4 つのサブタスク T1 ~ T4 を簡単に遂行することのできる次の 4 つの機能 F1 ~ F4 を提供している . 但し、F1 ~ F4 以外の機能を使用して T1 ~ T4 を遂行することは可能であるが、一般に、作業時間と共に実行機能数と機能実行回数が増加する . また、いわゆる文字入力作業だけで T1 ~ T4 を遂行することも可能である . 例えば、タスク T1 (目次の作成) は、機能 F1 (索引と目次) を使用しなくても、ユーザが章、節を列挙し、それらのページ番号を調べることにより、文字入力のみで遂行することが可能である . その場合、実行機能数と機能実行回数は減少するが、一般に、打鍵数等は大幅に増加する (T1 を遂行するためには、約 2000 文字の入力が余計に必要となる) ため、作業時間は (比較的大幅に) 増加することになる .

F1：「挿入」「索引と目次」...文書中の見出しの一覧を、指定した場所に挿入する。

F2：「挿入」「ページ番号」...文書のヘッダやフッタにページ番号を挿入する。

F3：「罫線」「変換」「文字列を表にする」...タブなどで区切られた文字列を表に変換する。

F4：「書式」「スタイル」...文書内の特定の文字列に対して、フォント、フォントサイズ、行間隔、文字配置等を一括して変更する。

ここで、カッコで囲んだ文字列は、これら機能に対応するメニュー項目名であり、矢印はメニュー階層を表している。なお、被験者には、これらサブタスク T1 ~ T4 と機能 F1 ~ F4 の存在は事前に知らされていない。

3.3 試作システム

試作システムの対象ソフトウェアは Microsoft Office 2000 である。図 13 は試作システムの概要を示したものである。ユーザの計算機上には Office 2000 とともに機能実行履歴収集部が稼働する。機能実行履歴収集部はユーザの操作を監視し、ユーザの機能実行履歴を機能実行履歴収集サーバへ送出する。ユーザの計算機と機能実行履歴収集サーバはイーサネットではばれている。

ユーザが通常どおり対象ソフトウェアを操作すると機能実行履歴が蓄積される。ユーザ操作のうち、メニュー選択とツールバー上の GUI ボタンの押下による機能実行が機能実行履歴として収集される。収集された機能実行履歴は、自動的に機能実行履歴収集サーバへ送出、蓄積される。ユーザは必要に応じて、機能実行履歴提示部から他ユーザ機能実行パターンを参照することができる。機能実行履歴そのものを参照することも可能である。

機能実行履歴収集部はプラグイン方式で実装した。機能実行履歴収集部は、Visual Basic で記述され 150 行程度の規模である。メニュー選択、ツールバー上の GUI ボタンの押下による全ての機能実行の度に、その内容がプラグインに通知される。ユーザの機能実行が通知されると、機能実行履歴収集部は、機能実行履歴を一時ファイルに蓄積し、一定量に達すると、実行履歴収集サーバへ送出する。

機能実行履歴受信部は、実行履歴収集サーバに常駐し、機能実行履歴収集部から送出された実行履歴をソフトウェア毎、ユーザ毎に階層化し、テキストファイルとして蓄積

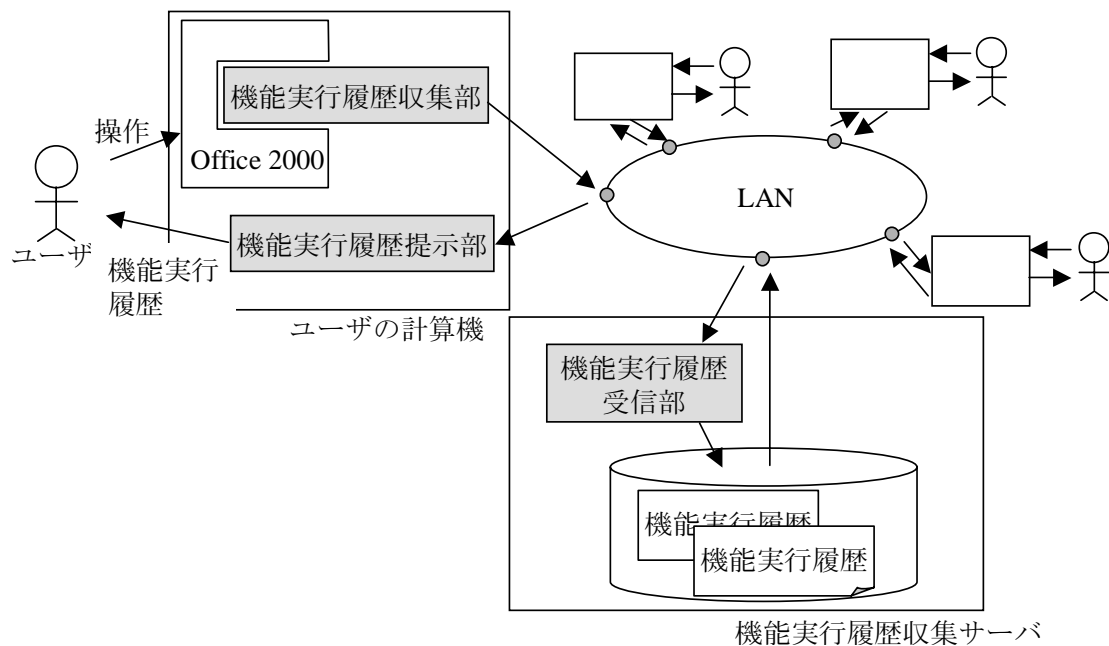


図 13 試作システムの構成

しておく。これらのファイルは OS が提供するファイル共有機能を利用して、各ユーザの計算機から読取り可能である。

機能実行履歴提示部は、機能実行履歴収集サーバに蓄積された各ユーザの機能実行履歴を読み込み、長さ 8 以下の機能実行履歴の部分系列を抽出するとともに、ユーザ毎の出現頻度を算出し、ユーザに提示する。部分系列の抽出の詳細は付録に添付した。

図 14は機能実行履歴提示部のスクリーンショットで、ユーザへの出力画面である。画面中央の表の各行が、提示部で抽出された機能実行履歴の部分系列を表す。各部分系列には次の項目が付加されている。

- 出現頻度: 全てのユーザの機能実行履歴における出現頻度
- 部分系列の長さ: 部分系列を構成する機能数
- ユーザ毎の出現頻度: カッコ内は頻度に対する割合

表のヘッダ部分をクリックすることで、ユーザは、これらの項目をキーとする機能実行履歴の部分系列の並べ替えを行なうことができる。例えば、他のユーザと比較して自

頻度	シーケンスの長さ	user 1 の頻度(%)	user 2 の頻度	user 3 の頻度	user 4 の頻度	user 5 の頻度	1	2	3
1	1	0 (0%)	0 (0%)	1 (100%)	0 (0%)	0 (0%)	太字(Alt)		
6	1	0 (0%)	0 (0%)	6 (100%)	0 (0%)	0 (0%)	段落書きと段落番号(Alt)		
1	1	0 (0%)	0 (0%)	1 (100%)	0 (0%)	0 (0%)	コピー(Alt)		
2	1	0 (0%)	0 (0%)	2 (100%)	0 (0%)	0 (0%)	貼り付け(Alt)		
2	1	0 (0%)	0 (0%)	2 (100%)	0 (0%)	0 (0%)	貼り付け(Alt)		
11	1	2 (27%)	8 (72%)	0 (0%)	0 (0%)	0 (0%)	索引と目次(Alt)		
2	1	0 (0%)	2 (100%)	0 (0%)	0 (0%)	0 (0%)	形式を選択して貼り...		
2	1	0 (0%)	2 (100%)	0 (0%)	0 (0%)	0 (0%)	文字列を赤にする(Alt)		
2	1	0 (0%)	2 (100%)	0 (0%)	0 (0%)	0 (0%)	セルの結合(Alt)		
2	1	0 (0%)	2 (100%)	0 (0%)	0 (0%)	0 (0%)	印刷(Alt)		
1	1	0 (0%)	1 (100%)	0 (0%)	0 (0%)	0 (0%)	すべて選択(Alt)		
2	1	1 (33%)	2 (66%)	0 (0%)	0 (0%)	0 (0%)	Microsoft Word ヘル...		
1	1	0 (0%)	1 (100%)	0 (0%)	0 (0%)	0 (0%)	元に戻す(Alt)自動ス...		
5	1	0 (0%)	5 (100%)	0 (0%)	0 (0%)	0 (0%)	元に戻す(Alt)書式設...		
1	1	0 (0%)	1 (100%)	0 (0%)	0 (0%)	0 (0%)	フィールド(Alt)		
1	1	0 (0%)	1 (100%)	0 (0%)	0 (0%)	0 (0%)	ヘッダーとフッター(Alt)		
1	1	0 (0%)	1 (100%)	0 (0%)	0 (0%)	0 (0%)	切り取り(Alt)		
1	1	0 (0%)	1 (100%)	0 (0%)	0 (0%)	0 (0%)	ページ番号(Alt)		
1	1	0 (0%)	1 (100%)	0 (0%)	0 (0%)	0 (0%)	閉じる(Alt)		
1	1	1 (100%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	操作履歴収集		
9	2	6 (66%)	0 (0%)	2 (22%)	0 (0%)	1 (11%)	スタイル(Alt)...	スタイル(Alt)...	
1	2	0 (0%)	0 (0%)	0 (0%)	0 (0%)	1 (100%)	スタイル(Alt)...	右揃え(Alt)	
1	2	1 (100%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	スタイル(Alt)...	フォント(Alt)	
10	2	2 (20%)	1 (10%)	2 (20%)	0 (0%)	4 (40%)	スタイル(Alt)...	スタイル(Alt)...	
1	2	0 (0%)	0 (0%)	0 (0%)	1 (100%)	0 (0%)	スタイル(Alt)...	ページ番号(Alt)	
1	2	0 (0%)	0 (0%)	1 (100%)	0 (0%)	0 (0%)	スタイル(Alt)...	コピー(Alt)	
1	2	1 (100%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	スタイル(Alt)...	索引と目次(Alt)	

図 14 試作システムの機能実行履歴提示部の画面例

らの実行頻度が少ない，あるいは，実行したことのない部分系列をユーザが知りたければ，自らの機能実行履歴における出現頻度をキーとして部分系列を昇順に並べ替えればよい．他にも，画面に表示させる部分系列の長さの最大値や出現頻度の最低値をユーザは指定可能である．

また，ユーザは機能実行履歴そのものをテキストエディタ等で閲覧することも可能である．機能実行履歴には機能実行毎にその「実行日」と「実行時刻」が記録されており，機能の実行時期をユーザは容易に知ることができる．

3.4 実験手順

実験は次の三つのフェーズで構成されている．

フェーズ 1： タスクの具体的内容が指示された文書(出来上がり見本のハードコピー)を閲覧してもらった上で、タスクを個別に実行してもらう。閲覧とタスク実行を合わせた作業時間は 30 分とした。なお、この 30 分という作業時間は、当該タスクの完了に必要なと思われる平均的な時間の半分となるよう、予備実験の結果に基づいて定めた。なお、被験者に対しては、次のような指示が出される。

- Microsoft Word 2000 の既知の機能を出来るだけ利用し、効率よくタスクを実行するように。
- 必要があればオンラインヘルプを参照しても構わない。
- 実行が困難と思われる作業は後回しにしても構わない。また、被験者の機能実行履歴は、試作システムによりリアルタイムに収集、蓄積される。

フェーズ 2： 被験者に、試作システムによる機能実行履歴提示画面を閲覧してもらう。提示画面には、被験者自身の機能実行履歴の部分系列やその頻度と共に、他の被験者の機能実行履歴の部分系列やその頻度も含まれている。それらを比較することにより、タスク実行に役に立ちそうな機能の利用知識の獲得を試みってもらう。なお、必要であれば、他の被験者の機能実行履歴そのものを被験者は閲覧することができる。閲覧時間の制約は特に設けない。

フェーズ 3： 被験者にタスク実行を再開してもらい、その終了まで作業を続けてもらう。ただし、ある被験者がタスクを完了できない、あるいは、既にタスクを完了した被験者と比べて非常に多くの時間をタスク完了までに要する、と考えられる場合、作業継続の意思の有無を本人に確認の上、タスク未完了ながら作業を終了させることもあり得る。なお、必要であれば、フェーズ 2 で閲覧した機能実行履歴やその分析結果を引き続き閲覧することが出来る。








3.5 結果

試作システムの機能実行履歴収集による計算機への負荷は非常に小さく、被験者のタスク実行を妨げることはなかった。

被験者ごとのフェーズ 1 とフェーズ 3 における実行機能数、機能実行回数を表 2 に示す。ここで、実行機能数は、タスクを実行する際に使用した機能の種類である。また、機能実行回数は機能実行履歴の総行数である。

表 1 被験者の Word 使用経験

被験者	作成ドキュメント	使用年数
S1	レポート, 論文 1, 卒論	1 年
S2	レポート, 案内状	2 年
S3	論文 多数	3 年
S4	レポート, 案内状, 論文 1	3 年
S5	論文 多数	5 年

なお、機能実行履歴を被験者（ユーザ）間で共有するフェーズ 2 の所要時間は被験者毎に異なるが、おおよそ 5 分から 10 分であった。また、4 つのサブタスクの遂行に役立つ 4 つの機能について、各被験者がどのフェーズでその利用知識を獲得し、タスク遂行に利用したかを図 15 に示す。図 15 において、 印は CLAS 方式による利用知識の獲得を、 印はオンラインヘルプによる利用知識の獲得を、 印は機能の実行を、それぞれ表す。 印のみの機能は、被験者にとってその利用知識が既知であったことを表す。また、 印、 印、 印のいずれもない機能は、利用知識の獲得もタスク遂行のための実行もなされなかったことを表す。

- 被験者 S4 はサブタスク T2（ページ番号の付加）を正しく終えることができなかった。なお、機能実行履歴を閲覧するフェーズ 2 の所要時間は、5 ～ 10 分であり、オンラインヘルプの一般的な参照時間等と同程度であった。
- 実験中に利用知識が獲得された機能はのべ 9 個で、そのうち、他の被験者の機能実行履歴を参照することで利用知識が獲得された機能はのべ 8 個、既存のオンラインヘルプ機能で利用知識が獲得された機能は 1 個のみであった。なお、試作システムを用いて他の被験者の機能実行履歴を参照することで利用知識が獲得された機能には、予め想定した機能 F1 ～ F4 以外の機能 F5（箇条書きと段落番号）が含まれている（被験者 S2 と S3 が利用知識を獲得した）。F5 を利用するとサブタスク T4 の遂行に必要な打鍵数は減少する。F4 を利用する方が、より簡単に T4 を遂行できるが、F1 ～ F4 と同様に「使用すると T1 ～ T4 を簡単に遂行できる」機能として、利用知識の獲得対象とみなした。

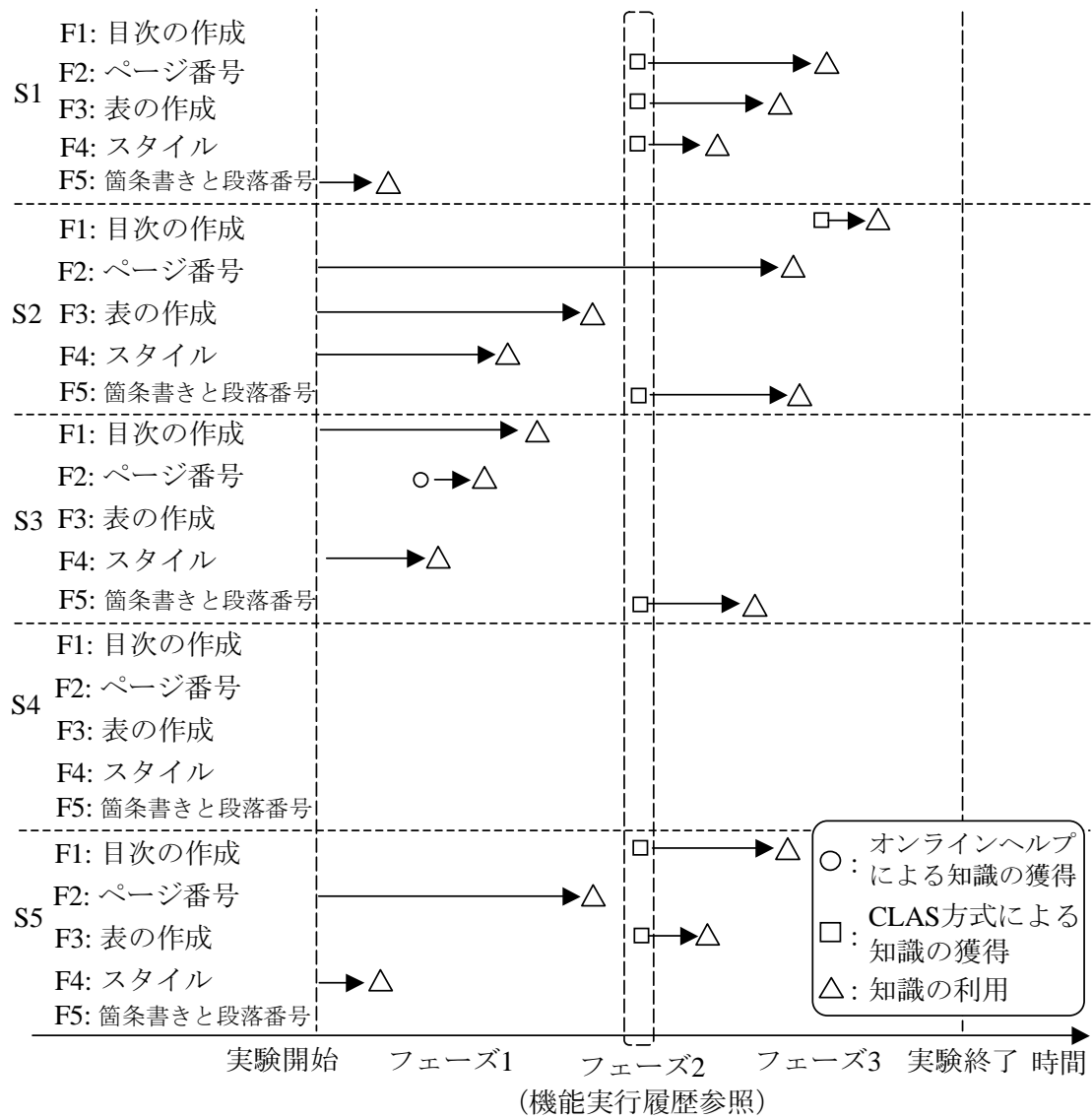


図 15 各被験者の利用知識の獲得時期

表 2 被験者の作業時間，実行機能数，機能実行回数

被験者	作業時間（分）				実行機能数	機能実行回数
	フェーズ 1	フェーズ 2	フェーズ 3	合計	合計	合計
S1	23	10	35	68	24	99
S2	24	6	52	82	21	134
S3	24	7	27	58	36	139
S4	23	9	*50	*82	21	167
S5	24	5	18	47	27	87

* サブタスク T2 未完了

表 2 と図 15 の情報，更に，実験後に被験者に対して行ったアンケートとインタビューの結果に基づいて，個々の被験者の特徴を簡単にまとめると次のようになる。

被験者 S1：5 名の被験者の中で最も初心者であり，4 つの機能 F1 ~ F4 いずれの利用知識も持っていなかった。しかし，フェーズ 2 において他の被験者の機能実行履歴を参照することで，3 つの機能 F2，F3，F4 の利用知識を獲得し，フェーズ 3 においてサブタスク T2，T3，T4 遂行のためにそれらを実行している。なお，機能 F2 が存在することを被験者 S1 は想像していたが，その具体的な実行方法は既知ではなかった。また，機能と F3，F4 については，その存在を想像さえしていなかった。

被験者 S2：3 つの機能 F2，F3，F4 の利用知識を持っていた。機能 F1 についてはその存在は想像していたが，他の被験者の機能実行履歴を参照することで，フェーズ 3 において具体的な実行方法（利用知識）を獲得している。

被験者 S3：2 つの機能 F1，F4 の利用知識を持っていた。機能 F2 については，フェーズ 1 においてオンラインヘルプを参照することで，その利用知識を獲得し，サブタスク T2 遂行のために実行している。また，予め想定した 4 つの機能 F1 ~ F4 とは別の「箇条書きと段落番号」機能の利用知識を，フェーズ 2 において他の被験者の機能実行履歴を参照することで獲得している。但し，機能 F3 については，オンラインヘルプによっても，また，フェーズ 2 において他の被験者の機能実行履歴を参照することによっても，その利用知識は獲得出来ていない。

被験者 S4：旧バージョンを含め，当該ソフトウェアの使用年数が 3 年と比較的に長いに

も関わらず、初心者の S1 同様、4 つの機能 F1 ~ F4 いずれの利用知識も持っていなかった。オンラインヘルプによっても、また、フェーズ 2 において他の被験者の機能実行履歴を参照することによっても、それらの利用知識を獲得することはなく、実行も行っていない。

被験者 S5：旧バージョンを含め、当該ソフトウェアの使用年数が 5 年と 5 名の被験者の中で最も長い。2 つの機能 F2, F4 の利用知識しか持っていなかった。但し、フェーズ 2 において他の被験者の機能実行履歴を参照することで、残りの 2 つの機能 F1, F3 の利用知識を獲得し、サブタスク T1, T3 遂行のためにフェーズ 3 でそれらを実行している。なお、被験者 S5 にとって、機能 F1 は具体的な実行方法が既知でなかっただけであるが、機能 F3 はその存在を想像すらしていなかった。4 つの機能 F1 ~ F4 を全て利用してサブタスクを実施した唯一の被験者である。

3.6 考察

評価実験により、CLAS 方式が妥当であることを示唆する結果が得られた。あるユーザがコミュニティ内の他のユーザの機能実行履歴を参照することにより、未知の機能を発見することができ、発見した機能による作業効率の向上につながることを確認できた。以下に評価実験で得られた知見を述べる。

- 初心者（被験者 S1）であっても、未知機能の発見が可能であった。しかも、機能の発見により、その後の作業時間は短くなり、機能実行回数は熟練者（被験者 S5）と同程度まで少なくなった。CLAS 方式は、初心者の作業効率の向上に寄与する可能性がある。
- ユーザが利用知識を獲得したのべ 8 個の機能のうち、ユーザがその存在を想像さえしていなかった機能がのべ 3 個もあった。そのうちの 1 つは、熟練者（被験者 S5）によって獲得されたものである。CLAS 方式が、初心者だけでなく熟練者に対しても有効である可能性がある。
- 提案方式の有効性をより明確に評価するために（被験者には知らされていなかったが）タスクには 4 つのサブタスクとそれらを行なうための 4 つの機能が予め設定（想定）されていた。しかし、それら 4 つの機能以外の機能についても、未知機能を発見した被験者が 2 名いた。このことは、タスク中に予め設定されていた機能のみが、CLAS 方式による未知機能の発見に特に適していたわけではないことを示

している。また、CLAS方式を実際利用する場合に当てはめてみると、ユーザに発見させる未知機能を予め明らかにしタスクを設定する、という必要が必ずしもないことになる。CLAS方式は、適用範囲が広く、実施コストも比較的小さい可能性がある。

- 機能実行履歴には、実行されたメニュー項目名だけでなく、メニュー階層内での位置が分かるよう、ルートメニューから実行されたメニュー項目名までの選択経路が付加されている。タスク完了後の被験者へのインタビューでは、こうした情報が、機能の試用やオンラインヘルプの利用において非常に役立った、との感想を得た。即ち、履歴中のある機能を試用したければ、ユーザは示された選択経路に従って実際にメニュー項目を選択するだけでよく、オンラインヘルプ上での検索では、表示されたメニュー項目名が非常に適切なキーワードとなった。

一方、5名の被験者のうち1名(被験者S4)は、CLAS方式による利用知識の獲得は見られなかった。提案手法の有効性や適用範囲に制限のある可能性もある。未知機能が発見できなかった主な原因と考えられる対策は次のとおりである。

- 被験者S4は、4つのサブタスクのうちT3(表の作成)の遂行に作業時間のほとんどを費やし、他のサブタスクは満足に終えることも出来なかった。与えられたタスクがユーザにとって負荷の高いものである場合、未知機能発見のためのよい動機付けになる反面、発見のための時間的、精神的余裕が生まれにくい可能性もある。ソフトウェアに対するユーザの熟練度、機能面での興味、等を勘案し、ユーザにとって無理のないタスク設定が必要と考えられる。
- 被験者S4にとって、試作システムが提示する機能実行履歴は膨大なものであった。試作システムに使い慣れていないこともあり、5～10分という比較的短い閲覧時間では、未知機能を見出すまでには至らなかった。同一タスク方式による未知機能発見であっても、タスク内容やユーザ数によって、機能実行履歴は非常に大きくなる可能性がある。膨大な機能実行履歴をコンパクトに提示する方式、ユーザの興味に合わせて提示内容を変更する(フィルタリングする)方式、等についてより詳細な議論が必要と考えられる。

また、評価実験により試作システムに関する以下のような問題が明らかになった。

- 機能実行履歴提示部の利便性

機能実行履歴の提示は短時間で済むべきものである。対象ソフトウェアを使用して本来の作業をしている合間に他ユーザの機能実行履歴を参照し、未知機能を発見することが望ましいが、試作システムでは、以下のような点で、機能実行履歴の参照に時間がかかり、評価実験に参加した被験者からも問題点として指摘された。

- 機能実行履歴提示部の処理速度

コミュニティ内の全てのユーザの機能実行履歴から一定長の部分系列を抽出するのに時間がかかった。評価実験では、5人のユーザの機能実行履歴はのべ626行であったが、部分系列の抽出には2、3分程度時間がかかった。長大な機能実行履歴を収集した場合には、処理時間が問題となる可能性がある。

- 機能実行履歴提示部の起動

試作システムでは、他ユーザの履歴を参照するために、以下のステップが必要であったが、煩わしいという意見が被験者から得られた。

1. 機能実行履歴提示部を起動する。
2. コミュニティ内の他ユーザの履歴ファイルをサーバからコピーする。
3. 機能実行履歴提示部からコピーしたファイルを指定し、部分系列を抽出する。

- 提示する情報

試作システムで提示される部分系列の数が多過ぎて未知機能の発見に至らない、もしくは、困難である、という意見が得られた。

- 提示される部分系列の数が非常に多い。

機能実行履歴に含まれるすべての部分系列では多過ぎる。試作システムの機能実行履歴提示部では、提示する部分系列の最大長を設定できたが、最大長を小さくしても、なお提示される部分系列の数が多く、という意見が得られた。

- 部分系列とユーザ毎の実行頻度だけでは不足である。

試作システムの機能実行履歴提示部では、機能実行履歴の部分系列とユーザ毎の機能実行履歴におけるその出現頻度を提示した。被験者から、ユーザは提示されたユーザ毎の出現頻度を機能毎に比較するのは面倒である、という意見が得られた。

- プライバシ

被験者へのインタビューから試作システムではプライバシーの保護が十分でないため、機能実行履歴収集に不安を感じる、という意見が得られた。具体的には、試作システムの機能実行履歴提示部を使用する際、他ユーザの機能実行履歴そのものを機能実行履歴サーバからダウンロードする必要があった。そのため、機能実行履歴ファイルに記された全ての内容が他ユーザから参照可能となる。例えば、機能の実行日時が他ユーザに知られてしまうと、ソフトウェアを使用していた期間がわかるためいやである、という意見が得られた。

4. 利用知識共有支援システム

4.1 システムの要件

3章の評価実験により，試作システムの問題点が明らかになった．これらの問題点を踏まえたシステムの要件は次のとおりである．

(1) 即応性

機能実行履歴送出力部，及び，提示部はユーザの指示により，素早く起動，実行する必要がある．そのため，システムでは次の点が満たされる必要がある．

(1-a) 機能実行履歴提示部起動の手間が小さい．

試作システムでは，機能実行履歴提示部は対象ソフトウェアとは別のソフトウェアとして，起動する必要があった．別のソフトウェアとして起動することは，ユーザに煩わしい操作を要求するため，この手間を小さくする必要がある．

(1-b) 部分系列抽出にかかる時間が短い．

試作システムでは，部分系列の抽出に時間がかかった．部分系列の抽出を短時間で済ませるためには，部分系列をあらかじめ抽出しておくなどの処理が必要である．

(2) 機能実行履歴提示部の提示方法

(2-a) 提示する部分系列の数を小さくする．

試作システムで提示される部分系列の数が多過ぎて未知機能の発見に至らない，もしくは，困難である，という意見が得られた．提示される機能実行履歴の部分系列に含まれる機能実行のコンテキストを失うことなく，文献 [41][45][42] のように膨大な量の情報を要約し，コンパクトに提示する方法が必要である．

(2-b) 付加情報

部分系列とユーザ毎の出現頻度だけでなく，文献 [5] のようにユーザにとってその機能がどの程度重要であるかを示す情報を付加するなどによりユーザを積極的に支援する必要がある．

(3) プライバシの保護

被験者へのインタビューから試作システムではプライバシーの保護が十分でないため、機能実行履歴収集に不安を感じる、という意見が得られた。コミュニティ内のユーザからは、直接機能実行履歴を参照することができなくする必要がある。

(4) 広範囲な履歴の収集

試作システムでは、機能実行履歴の収集を HTTP (Hyper Text Transfer Protocol) を通じて行った。広範囲な機能実行履歴の収集には、HTTP を使用することが適していることがわかった。機能実行履歴の収集には、特別な装置が必要ないだけでなく、ファイアウォールなどの制限を受けにくい HTTP を使用することが望ましい。また、試作システムでは履歴の収集の際にしか HTTP を使用しなかったが、履歴の参照の際にも HTTP が使えることが望ましい。

4.2 構成

要件 (1-a) を満たすために、機能実行履歴提示部は対象ソフトウェアに統合した。要件 (1-b) を満たすために、ユーザの計算機から機能実行履歴を送出するタイミングを対象ソフトウェア終了後とし、送出了た機能実行履歴を受信した機能実行履歴サーバは受信直後に、ユーザ毎に部分系列をあらかじめ抽出し、その出現頻度をあらかじめ計数しておき、ユーザから要求があった場合には、ユーザ毎の部分系列をマージして送出すことにした。

要件 (2-a) を満たすために、機能実行履歴の提示を部分系列一覧提示部と実行頻度一覧提示部に分割した。部分系列一覧提示部では、履歴を参照するユーザが指定した機能を含む部分系列だけを提示する。実行頻度一覧提示部では、各ユーザの機能実行履歴に含まれる単一の機能とその実行頻度を提示する。また、要件 (2-b) を満たすために、実行頻度一覧提示部では、履歴を参照するユーザの実行頻度がゼロである機能を未知の機能とし、強調表示する。また、部分系列の一覧では、共起、連鎖する機能を視覚的に捉えることができるように、部分系列を構成する各機能を機能名毎に色分けして提示できるようにする。

要件 (3) を満たすために、機能実行履歴そのものはサーバで保管し、外部からは参照不可とした。要件 (1-b) と関連するが、履歴を参照しようとするユーザには、機能実行履歴の部分系列のみが送信されるため、機能の実行日時などの情報が送信されることはな

い。同時に、サーバから送出する部分系列に含まれるユーザ名を匿名とすることも可能である。

要件(4)を満たすために、機能実行履歴の送受信には HTTP を使用した。機能実行履歴の収集、及び、参照、共に HTTP を使用するため、広範囲な履歴の収集が可能であるだけでなく、対象ソフトウェアや機能実行履歴サーバのプラットフォームや OS に依存しない形式で機能実行履歴の収集、参照が可能である。また、HTTPS (HTTP over SSL) を使用すれば、即座に秘匿性の高い通信が可能になる。

システムの構成を図 16 に示す。システムはソフトウェアを実行する各ユーザの計算機上で実行されるクライアント、機能実行履歴を蓄積する一つのサーバ、及び、それらを結ぶネットワークで構成される。要件(4)を満たすためにクライアント・サーバ間の通信はすべて HTTP (Hyper Text Transfer Protocol) を使用する。HTTP を使用することにより、対象ソフトウェアの記述言語や OS に依存することなく機能実行履歴を集めることができる。対象ソフトウェアは Microsoft Office 2000 を選んだ。Office 2000 アプリケーションである Word, Excel, PowerPoint に用意されたメニュー項目数は contextual メニュー²を含めそれぞれ、約 500, 800, 400 であり、膨大な数の機能を提供するソフトウェアであるといえる。

クライアントでは、Office 2000 アプリケーション群のいずれか、機能実行履歴収集部、機能実行履歴提示部が実行される。機能実行履歴収集部は、2章で述べた収集方式のプラグイン方式で実装した。要件(1-a)を満たすために機能実行履歴提示部は Office 2000 のプラグインモジュール (COM add-in) として実装した。機能実行履歴提示部、及び、ユーザ選択部は、Office のメニュー選択により、起動できる。クライアントの各部は Visual Basic で記述し、合計 3,000 行程度の規模である。

機能実行履歴サーバでは、機能実行履歴の蓄積のために、機能実行履歴受信部、部分系列抽出部を実行する。また、機能実行履歴の参照のために部分系列送出部、及び、ユーザ属性送出部を実行する。サーバで実行する各部は Java Servlet として実装され、HTTPD (HTTP Daemon) と共に常駐し、クライアントからの要求を受付ける。サーバの各部は Java 言語で合計 13,000 行程度の規模である。

² マウスの右クリックで現れるメニュー

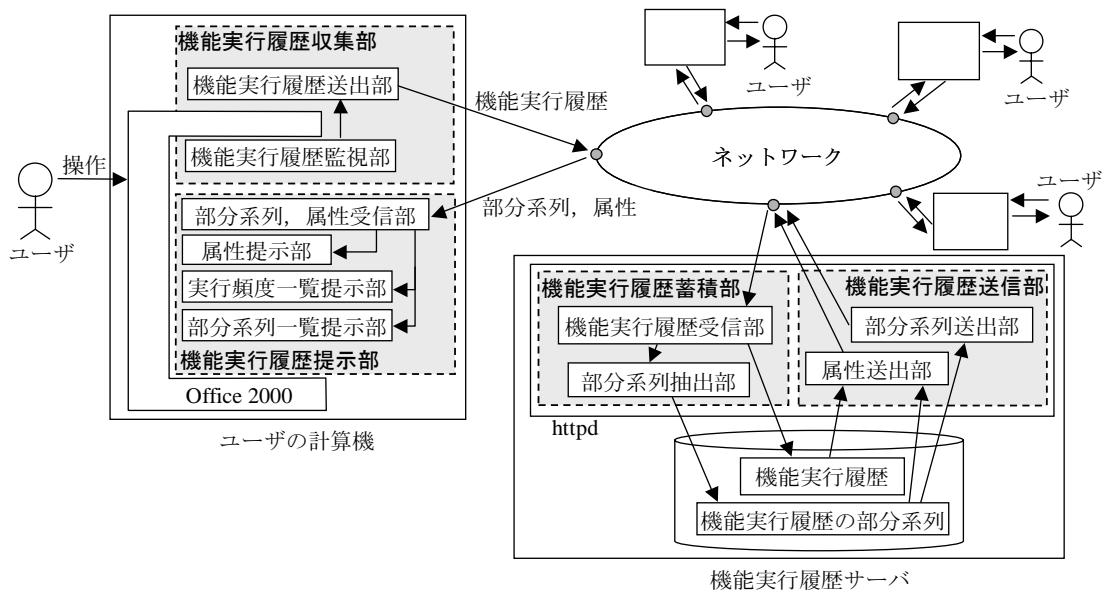


図 16 システムの構成

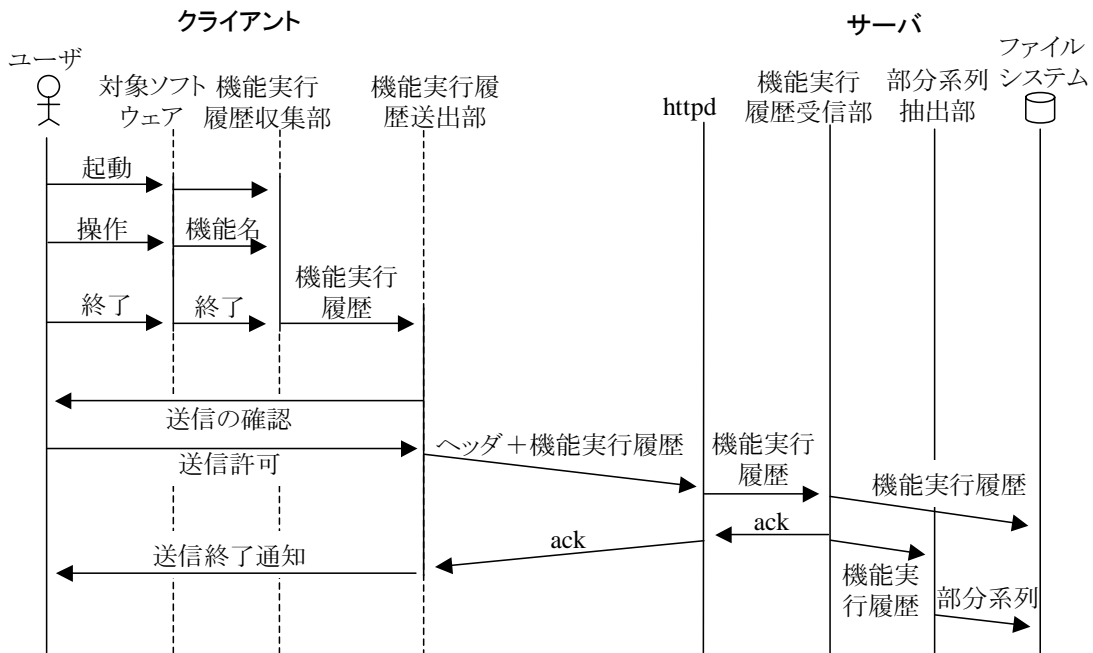


図 17 機能実行履歴収集のデータの流れ

2000/02/03 18:50:41 書式設定->フォント (&F)
2000/02/03 18:50:45 書式設定->フォント サイズ (&F)
2000/02/03 18:50:48 標準->中央揃え (&C)
2000/02/03 18:51:16 ファイル->上書き保存 (&S)
2000/02/03 18:51:23 ファイル->終了 (&X)

図 18 機能実行履歴の例

4.3 機能実行履歴の収集

図 17に機能実行履歴収集時のデータの流れを示す。図 17中の矢印はデータの流れ、矢印についたラベルはデータの内容を表す。機能実行履歴収集部は、Office 2000 と共に起動する。収集部はユーザの Office 2000 に対するメニュー選択、メニュー選択と同等の結果を得られるショートカットボタンの押下の度に、メニューのキャプション、ショートカットボタン名とその選択経路に実行日時を付加し、機能実行履歴としてクライアント上のファイルに蓄積する。図 18に機能実行履歴の例を示す。図 18の例では、1つの行が1つの機能実行に対応し、機能実行毎に「実行日」、「実行時刻」、「メニュー項目名」が記録されている。メニュー項目名については、メニュー階層内での位置が分かるように、ルートメニューからの選択経路を付加する。例えば、図 18に示す履歴の先頭行の場合、選択されたメニュー項目「フォント」は、ルートメニューの「書式」のサブメニュー内に位置することが分かる。機能実行履歴収集部は、Office 2000 終了と共に終了される。その際、機能実行履歴収集部は機能実行履歴送出处を起動し、履歴を収集したファイルを渡す。

機能実行履歴送出处はファイルを受取り、ユーザに機能実行履歴を送信してよいかを尋ねる。ユーザが送信を禁止した場合、機能実行履歴送出处はファイルを削除した後、終了する。ユーザが送信を許可した場合、機能実行履歴送出处は機能実行履歴サーバの HTTPD と接続し、ユーザ名、ソフトウェア名が記されたヘッダを付加し、機能実行履歴を送出する。要件 (3) を満たすよう機能実行履歴送出处は、ユーザに送信の可否を求めるときに、ユーザが送信する機能実行履歴を閲覧するための機能を持つ。ユーザは、機能実行履歴を送信する前にどのような情報が送信されるかがわかり、送信の可否を決めることができる。

機能実行履歴送出处が機能実行履歴を送信すると、HTTPD 経由で機能実行履歴受信

部が呼出される。機能実行履歴受信部は、機能実行履歴送出部が送信した機能実行履歴のヘッダからソフトウェア、ユーザ名を取得し、機能実行履歴をソフトウェア、ユーザ名により分類した上で、蓄積し、部分系列抽出部を呼出す。機能実行履歴受信部は、部分系列抽出部を呼出した後、機能実行履歴送出に受信が済んだことを知らせる。

部分系列抽出部は、履歴を送信したユーザの機能実行履歴とこれまでに蓄積した機能実行履歴とに含まれる部分系列を抽出し、長さ 6 以下の部分系列をファイルに出力する。部分系列の最大長である 6 は、3章における評価実験において、実験後の被験者の感想から得た「コンテキストを理解するために十分な部分系列の長さ」の平均値である。

要件 (1-b) を満たすよう、機能実行履歴受信と同時に部分系列を抽出しておくことにした。あらかじめユーザ毎の部分系列を抽出しておくことにより、機能実行履歴サーバから機能実行履歴を送出する際の処理時間を短くすることができる。送出時には部分系列抽出部が抽出したユーザ毎の部分系列をマージして送出する。

4.4 機能実行履歴の提示

機能実行履歴提示部は部分系列・属性受信部、属性提示部、実行頻度一覧提示部、部分系列一覧提示部から構成される(図 16)。図 21、図 27、図 28は、それぞれ、機能毎の実行頻度一覧、部分系列の一覧、機能実行履歴の属性を提示する画面例である。本節では、これらの画面について述べた後、機能実行履歴サーバとクライアント間のデータの流れについて述べる。

4.4.1 提示画面

要件 (2) を満たすために、2つの提示画面を用意した。1つは機能毎の実行頻度一覧であり、有用な未知機能の発見を支援する。機能毎の実行頻度一覧で提示する機能は、長さ 1 の部分系列(単一機能)であり、3章で述べた試作システムが提示する部分系列の数と比較すると提示される機能の数は極めて小さい。

もう1つの提示画面は機能実行履歴の部分系列の一覧である。機能実行履歴の部分系列の一覧では、長さ 2 以上の部分系列のうち、ユーザが指定した機能を含む部分系列だけを提示する。ユーザが指定した機能を含む部分系列だけに限定することにより、3章で述べた試作システムが提示する部分系列の数と比較すると提示される機能の数は大幅に小さくなる。

- 機能毎の実行頻度一覧



図 19 機能毎の実行頻度の提示方法

機能毎の実行頻度一覧は、機能実行履歴を参照するユーザにとって未知の機能を他ユーザの実行履歴から発見することを支援する。一覧は、図 19 のように、コミュニティに属するユーザの機能実行履歴に含まれる機能名とその頻度を算出した上で、機能名と実行頻度を対にして提示する。

また、要件 (2-b) を満たすよう、以下によりユーザが未知の機能を発見することを支援する。

- 実行頻度がゼロである機能の強調表示
 - 一覧を参照するユーザの実行頻度がゼロである機能は、そのユーザにとって未知の機能であるとみなし、強調表示する。ユーザは実行頻度がゼロである

機能名	実行頻度		
	ユーザA	ユーザB	ユーザC
機能A	1	3	0
機能B	0	2	0
機能C	2	1	1
機能D	2	0	3
機能E	1	0	2

他ユーザによる実行率	実行頻度				機能名
	全ユーザ	ユーザA	ユーザB	ユーザC	
50%	4	1	3	0	機能A
0%	2	0	2	0	機能B
100%	3	2	1	1	機能C
100%	2	2	0	3	機能D
100%	1	1	0	2	機能E

図 20 機能毎の実行頻度の提示方法

機能を見つけることが容易になる。図 20 の例では、一覧を参照するユーザがユーザ B であるとし、ユーザ B の実行頻度がゼロである機能 D、及び、機能 E を強調表示している。

– 他ユーザによる実行率

一覧を参照するユーザ以外のユーザの何割がその機能を実行したかを機能毎に算出し、機能毎に提示する。ユーザは、提示されている各機能に対しそのユーザ以外の何人のユーザがその機能を実行しているか分かる。図 20 の例では、一覧を参照するユーザがユーザ B であるとし、機能 A では残り二人のユーザのうち、一人が実行しているため、他ユーザによる実行率が 50%となっている。機能 B はユーザ B だけが実行しているため、他ユーザによる実行率が 0%となっている。

– 機能を実行するための操作

提示する機能名にルートメニューからの選択経路もしくはボタンバー（ツールバー）の名称を付加する。ユーザは機能が GUI メニュー、ボタンバー上のどこに属しているかが分かることにより、機能実行の再現（実際にその機能を実行すること）が可能となる。

– 指定した機能の前後に実行された機能の提示

一覧から 1 つの機能を選択して、次節で述べる機能実行履歴の部分系列の一覧をユーザが閲覧できるようにする。ユーザは一覧から発見した機能の実行コンテキストを理解したいと思った場合に、その機能を含む部分系列を閲覧することができる。ユーザは指定した機能の前後に実行される機能から、その機能を実行するコンテキストや指定した機能を実行するためにあらかじめ

実行回数	実行比率	user A	user B	user C	user D	user E	機能名
100%	26	2600	10000	5140	2000	9000	Cell Alignment(セルの配置)の均等揃え (中央)
100%	43	5020	7100	8100	7100	16000	Edit(編集)のクリア(A)
100%	52	5000	21400	6100	13000	7100	Standard(標準)のズーム(Alt)
100%	165	10100	21100	17100	15000	90000	Formatting(書式設定)の斜体(B)
100%	27	10000	14000	1400	1400	1400	Insert(挿入)の新ページ(B)
100%	270	1140	60000	1140	152000	36100	Formatting(書式設定)の太字(B)
100%	26	3020	11400	4000	6000	2000	Standard(標準)の上書き保存(B)
100%	115	17000	11100	41000	21000	25000	Tables(テーブル)の移動(B)
100%	17	2020	3100	3100	2100	7400	View(表示)のアウトライン(B)
100%	43	8000	7100	13000	13000	1000	File(ファイル)の終了(B)
100%	106	11000	24000	9000	49400	13100	Table(テーブル)のセルの結合(B)
100%	21	2100	7000	9000	3140	1000	Format(書式)のスタイル(B)
100%	190	1400	86400	20100	57000	15000	Insert(挿入)の表と目次(B)
100%	16	2010	1000	10000	4000	1000	Edit(編集)の検索(B)
75%	22	3040	6000	1000	12000	0000	File(ファイル)の上書き保存(B)
75%	9	0000	0000	10100	40400	40400	Clipboard(クリップボード)のクリップボード
75%	5	10000	0000	10000	0000	10000	Drawings(図形)の印刷(B)
75%	21	4000	1000	15000	1000	0000	Format(書式)の段落(B)
75%	32	0000	0000	60000	10000	0000	Standard(標準)の閉じる(B)
75%	13	10000	0000	20100	1000	0000	Formatting(書式設定)の右揃え(B)
75%	56	6010	21000	22000	7100	0000	ツールバーのツールバーの文字列
75%	56	4000	11000	24400	17000	0000	View(表示)の下書き(B)
75%	104	2000	0000	96000	6000	0000	Order(順序)のテキストの前面へ移動(B)
75%	42	2000	15000	23000	2000	0000	Insert(挿入)のフィールド(B)
75%	43	6040	1000	0000	20400	16000	Edit(編集)の形式を選択して貼り付け(B)
75%	15	4000	0000	10000	1000	0000	Formatting(書式設定)のフォント(B)
75%	7	14000	1140	2000	0000	1140	File(ファイル)の印刷(B)
75%	419	19000	0000	169000	31000	0000	Formatting(書式設定)の両端揃え(B)
75%	20	21000	21000	1000	15000	0000	Help(ヘルプ)のMicrosoft Word ヘルプ(B)
75%	106	10000	12110	53000	31000	0000	Table(テーブル)のテキストの移動(B)
75%	13	21000	21000	9000	1000	0000	Formatting(書式設定)のインデント(B)
75%	13	7000	8000	0000	2000	1000	Review(訂正)のオブジェクトの書式設定(B)

図 21 機能毎の実行頻度一覧の画面例

実行しておくべき機能等を知ることができる。

図 21 は 5 人のユーザ user A から user E までの Word 2000 の機能実行履歴を user B が参照した際の機能毎の実行頻度一覧の画面例である。図 21 中の表の各行は一つの機能に対応する。表の各列は、左から順番に、他ユーザによる実行率（1 回以上実行したことがあるユーザの割合）、実行回数（各ユーザの実行頻度の合計）、ユーザ毎の実行頻度、メニュー項目名、もしくは、ショートカットボタン名（機能名）である。図 21 中の右側のチェックボックスを操作することにより、メニュー、ショートカットボタンまでの選択経路の表示・非表示を切替えることができる。これにより、ユーザが機能を実行するための操作方法を理解することを支援する。機能名がダブルクリックされると、その機能を指定機能とした部分系列の一覧が提示される。

一覧表の各行は、表のヘッダ部分をクリックすることにより「機能名」以外の項目で昇順、降順に並べ替えることができる。ヘッダ部分を複数クリックすることによ

ユーザID	実行率	機能名
1000	10	File(ファイル)を開く(O)
1000	87	Insert(挿入)の表書き(O)
1000	22	File(ファイル)を開く(O)
1000	11	Format(書式)設定(O)
1000	12	Send To(送る)のメニューを開く(O)
1000	23	Format(書式)設定(O)
1000	24	Format(書式)設定(O)
1000	112	Table(表)のメニューを開く(O)
1000	30	Format(書式)設定(O)
1000	41	File(ファイル)を開く(O)
1000	23	Drawing(図面)のメニューを開く(O)
1000	45	Format(書式)設定(O)
1000	33	Standard(標準)のメニューを開く(O)
1000	83	Insert(挿入)のメニューを開く(O)
1000	136	Format(書式)設定(O)
000070	2	Shapes(図形)のメニューを開く(O)
000070	4	File(ファイル)を開く(O)
000070	2	Shapes(図形)のメニューを開く(O)
000070	13	Format(書式)設定(O)
000070	22	Insert(挿入)のメニューを開く(O)
000070	2	Format(書式)設定(O)
000070	4	File(ファイル)を開く(O)
000070	3	File(ファイル)を開く(O)
000070	3	Header and Footer(ヘッダとフッター)のメニューを開く(O)
000070	12	File(ファイル)を開く(O)
000070	2	Windows(ウィンドウ)のメニューを開く(O)
000070	5	Format(書式)設定(O)
000070	16	Format(書式)設定(O)
000070	5	File(ファイル)を開く(O)
000070	5	Insert(挿入)のメニューを開く(O)
000070	70	Print Preview(印刷プレビュー)のメニューを開く(O)

図 22 並べ替えの例 (第 1 キー: 他ユーザによる実行率 (降順), 第 2 キー: 履歴参照ユーザの実行頻度 (昇順))

り, Table Lens[34] と同様に, 並べ替えのキーを複数指定することができる. 例えば, ユーザが 1 列目のヘッダをクリックすると, 各行が 1 列目の値により並び替わる. 次に, 2 列目のヘッダをクリックすると, 並び替えられた各行のうち 1 列目の値が等しい行が 2 列目の値で並び替わる. ユーザはこれを利用して, 2.5 節で述べた 3 つの機能の集合を以下のように参照することができる.

- 他の多くのユーザが実行したことがあり, かつ履歴を参照するユーザの実行頻度が小さい機能
他ユーザによる実行率の降順で並べ替えた後, 履歴を参照するユーザの実行頻度の昇順で並べ替える. すると, 一覧表の上の方の行の機能がこの集合に含まれる (図 22 中の枠内).
- 履歴を参照するユーザしか実行したことがない機能
他ユーザの実行率の昇順で並べ替えた後, 履歴を参照するユーザの実行頻度の昇順で並べ替える. すると, 一覧表の上の方の行の機能がこの集合に含まれる (図 23 中の枠内)
- 他ユーザによる実行頻度が高く, かつ, 履歴を参照するユーザが実行したこ

実行項目	実行回数	実行ユーザ	実行時刻	実行内容
Tableの挿入	1	ODR0	0000	100000 Tableの挿入
Fileの挿入	1	ODR0	0000	100000 Fileの挿入
Insert	1	ODR0	0000	100000 Insert
Edit	1	ODR0	0000	100000 Edit
Fileの削除	1	ODR0	0000	100000 Fileの削除
Tableの削除	1	ODR0	0000	100000 Tableの削除
Whole Tableの削除	2	ODR0	0000	200000 Whole Tableの削除
Cell Alignment	4	ODR0	0000	400000 Cell Alignment
Print Preview	4	ODR0	0000	400000 Print Preview
Formatting	5	ODR0	0000	500000 Formatting
Tableの挿入	6	ODR0	0000	600000 Tableの挿入
Insert	7	ODR0	0000	700000 Insert
Table Text	8	ODR0	0000	800000 Table Text
Picture Menu	17	ODR0	0000	1700000 Picture Menu

図 23 並べ替えの例 (第 1 キー: 他ユーザによる実行率 (昇順), 第 2 キー: 履歴参照ユーザの実行頻度 (昇順))

とのない機能

各ユーザの実行頻度の合計の降順で並べ替えた後、履歴を参照するユーザの昇順で並べ替える。すると、一覧表の中の強調表示される機能がこの集合に含まれる (図 24 中の枠内)

- 機能実行履歴の部分系列の一覧

部分系列の一覧は、ユーザが指定した機能を実行するコンテキストを理解することを支援する。一覧は、機能実行履歴から部分系列を抽出し、長さ 2 以上の部分系列を KWIC (Key Word In Context) 形式で提示する。具体的には、実行履歴に含まれる部分系列を抽出し、その出現頻度を計数する。部分系列は図 25 のように、指定機能を中心として、その出現頻度と対して提示する。部分系列を提示することにより、指定機能を実行する前にあらかじめ実行しておく必要のある機能を知ることができる。また、指定機能と共に、連鎖する機能から指定機能がどのような機能であるか想像し易くなるのが期待できる。

また、要件 (2-b) を満たすよう、以下によりユーザが指定した機能を実行するコンテキストを理解することを支援する。

実行頻度	実行回数	実行回数	実行回数	実行回数	機能名	
30.33%	1811	25730	9200	9200	10455000	Fileメニューの「上書き保存」
30.33%	166	9200	80000	9200	1167200	Font menuの「設定」の「スタイル」
100%	126	9200	71000	9200	820000	Font menuの「設定」の「中央揃え」
100%	112	30070	80000	170500	7000	Toolsメニューの「文字入力」
99.97%	73	101000	9200	200000	304000	Standardメニューの「上書き保存」
100%	87	80200	610000	80200	80200	Insertメニューの「挿入」
0%	0	9200	9200	9200	600000	Font menuの「設定」の「フォント」
100%	53	1000	90700	9000	407000	Insertメニューの「挿入」
99.97%	51	60000	10000	9200	210000	Editメニューの「設定」の「フォント」
100%	45	104000	9000	60000	104000	Font menuの「設定」の「フォント」
100%	41	100000	110000	60000	140000	Fileメニューの「挿入」
99.97%	38	9200	200000	60000	100000	Fileメニューの「挿入」
33.33%	34	80000	1400000	80000	80000	Insertメニューの「挿入」
100%	32	60000	2200000	40000	80000	File menuの「設定」の「フォント」
100%	30	10000	60700	10000	110000	Font menuの「設定」の「フォント」
30.33%	29	9200	110000	9200	100000	Shapesメニューの「テキスト」
30.33%	25	9200	104000	9200	100000	Font menuの「設定」の「スタイル」
100%	24	60000	104000	10000	50000	Font menuの「設定」の「右揃え」
100%	23	70000	60000	50000	50000	Font menuの「設定」の「文字揃え」
100%	20	10000	9000	40000	100000	Drawingメニューの「テキスト」
99.97%	22	80000	1900000	80000	80000	Insertメニューの「挿入」
99.97%	20	9200	20000	110000	60000	Font menuの「設定」の「フォント」
30.33%	20	9200	1000	9200	100000	Tableメニューの「設定」
99.97%	17	10000	110000	10000	10000	Insertメニューの「挿入」
100%	17	60000	60000	20000	20000	Font menuの「設定」の「フォント」
0%	17	9200	9200	9200	170000	Picture menuの「設定」
99.97%	15	9200	100000	10000	20000	Font menuの「設定」
30.33%	15	9200	140000	9200	10000	Tables and Bordersメニューの「設定」
30.33%	15	9200	30000	9200	100000	Cell Alignmentメニューの「設定」
99.97%	15	80000	700000	80000	80000	Formatメニューの「設定」

図 24 並べ替えの例 (第 1 キー: 実行頻度の合計 (降順), 第 2 キー: 履歴参照ユーザの実行頻度 (昇順))

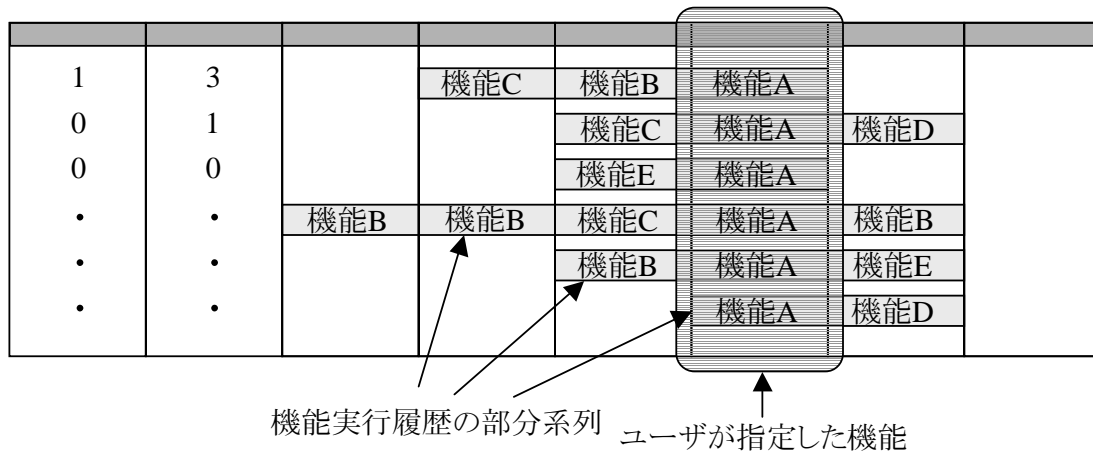


図 25 部分系列の提示方法

1 0 0 . . .	3 1 0 . . .		機能C	機能B	機能A	
				機能C	機能A	機能D
				機能E	機能A	
		機能B	機能B	機能C	機能A	機能B
				機能B	機能A	機能E
					機能A	機能D

↑
ユーザが指定した機能

図 26 部分系列に含まれる同一機能の色分け

- 一覧を参照するユーザがすでに実行したことがある部分系列の除去
 - 一覧を参照するユーザが 1 回以上実行したことがある部分系列は、ユーザが機能を実行するコンテキストを理解しているものと見なし、その部分系列をユーザの指定により省くことができる。
- 部分系列に含まれる同一機能の色分け
 - 指定した機能と共起、連鎖する機能を視覚的にわかりやすく提示するために、同一機能をユーザの指定により強調表示できる。図 26 は、ユーザが「機能 B」を指定し、部分系列中に含まれる機能 B を強調表示した例である。

図 27 は Word 2000 の機能実行履歴の部分系列の提示例である。図 27 中の表の各行は 1 つの部分系列に対応する。画面左側の他ユーザによる実行率、ユーザ毎の出現頻度を提示する部分は、機能毎の実行頻度一覧と同様である。画面右側は機能実行履歴の部分系列を提示している。図 27 では指定機能を「索引と目次」としている。機能毎の実行頻度一覧と同様にメニュー項目、ショートカットボタンまでの選択経路の表示・非表示を切替えることができる。表中の部分系列を構成する機能をクリックして選択することにより、表中のその機能と同じ機能が色づけされる。表中の機能が複数選択された場合には、選択機能毎に色分けして表示する。図 27 は、機能名「スタイル」、「図表番号」、「フィールド更新」を選択し、色分けした例である。

図 27 指定機能を中心とした部分系列一覧の画面例

「指定したユーザの部分系列除去」チェックボックスを操作することにより、履歴を参照するユーザの実行履歴に含まれる系列を一覧表から除去することができる。

- ユーザ選択画面

図 28は、ユーザ毎の機能実行履歴の属性一覧の画面例である。表の各行は一人のユーザの機能実行履歴に対応する。表の各列は、ユーザ名、1回以上実行したことがある機能の数（使用機能の種類）、使用機能のうち、履歴を参照するユーザの使用機能と共通する機能の数（共通する機能数）、機能実行の回数（機能実行履歴の総行数）、機能実行履歴送信の回数、初めて機能実行履歴を送信した日時（初回受信日時）である。「ユーザ属性をダウンロード」ボタンを押すことにより、サーバに蓄積されている同一ソフトウェアの機能実行履歴の属性をサーバに要求することができる。また、図 28中の右側のリストから履歴を参照しないユーザを選択することにより、特定のユーザの履歴を参照しないようにすることも可能である。デフォルトでは、機能実行履歴サーバに蓄積されたすべてのユーザの機能実行履歴を参照する。「履歴をダウンロード」ボタンを押すと、機能実行履歴をサーバに要求することができる。

4.4.2 データの流れ

部分系列・属性受信部は、属性提示部、実行頻度一覧提示部、部分系列一覧提示部からの要求を受けて、ユーザ毎の機能実行履歴の属性、及び、実行頻度つき部分系列を機

ユーザ名	実行履歴の種別	実行する履歴数	履歴の件数	履歴実行日時
akira-ma	43	23	236	5 Wed Jul 26 12:16:5
akira-m	12	34	376	14 Fri Jul 14 22:52:51
chibugan	4	4	11	7 Thu Aug 13 21:11:53
kenta-ko	7	5	5	2 Fri Aug 04 16:28:15
masata-y	92	28	651	15 Wed Jan 12 02:37:2
masatako	64	25	541	11 Tue Aug 18 16:22:1
shohei-m	18	12	84	2 Wed Mar 26 21:5:12
shout-i	24	12	191	2 Wed Mar 26 20:54:1
shohei-m	60	65	201	24 Sun Jan 29 13:06:5
hase-i	25	14	155	2 Thu Apr 21 20:31:5
hase-ko	23	13	125	2 Wed Mar 26 21:41:1
yama-i	28	18	258	2 Wed Mar 26 21:50:1

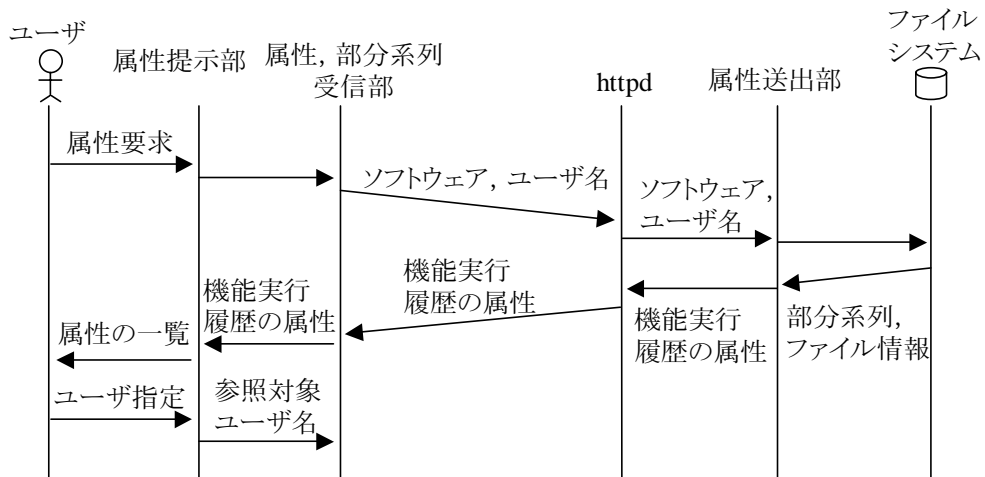
図 28 蓄積した機能実行履歴の属性一覧の画面例

能実行履歴サーバから受信する．部分系列・属性受信部は，受信した各ユーザの機能実行履歴の属性を属性提示部に，長さ 1 の部分系列（単一機能）を実行頻度一覧提示部に，長さ 2 以上の部分系列を部分系列提示部に，それぞれ渡す．

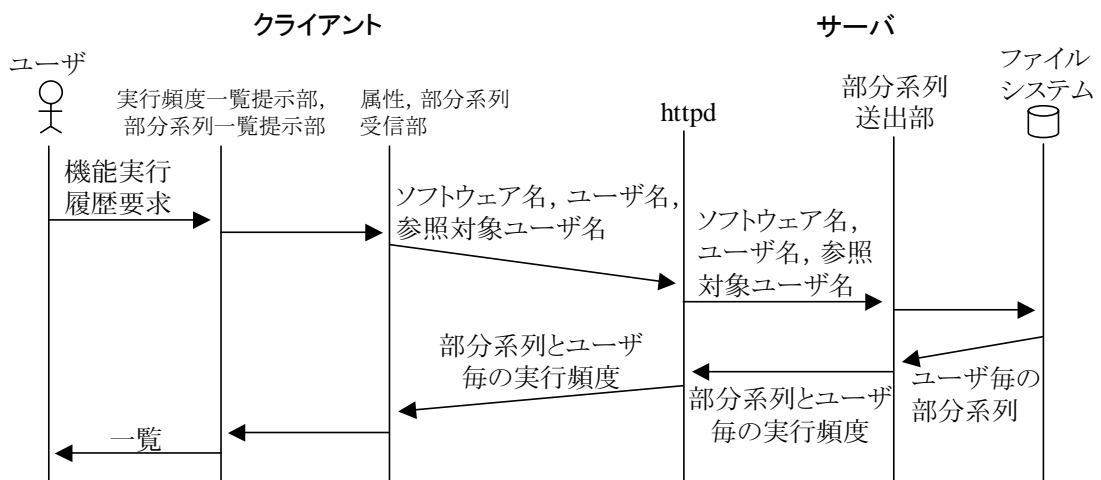
機能実行履歴を参照する際のデータの流れを図 29 に示す．図 17 と同様に，図中の矢印はデータの流れを，矢印についたラベルはデータの内容を表す．図 29(a) は属性・部分系列受信部が機能実行履歴の属性を機能実行履歴サーバに要求する際のデータの流れを表している．履歴を参照するユーザにより各ユーザの機能実行履歴の属性が要求されると，属性提示部を通じて，属性・部分系列受信部が通知を受ける．属性・部分系列受信部は，履歴を参照するユーザが起動しているソフトウェア名，及び，そのユーザ名を HTTPD 経由で属性送出部に送信する．

属性送出部は，ファイルシステムに蓄積されたユーザ毎の部分系列，及び，ファイルの数，作成日時から，3 章で述べた機能実行履歴の属性を作成し，HTTPD 経由で属性・部分系列受信部に送信する．属性送出部からの属性は，図 30 に示すような XML(eXtensible Markup Language) で記述されている．属性・部分系列受信部は，受信した属性をパースし属性提示部に渡す．属性を XML で記述することにより，属性の項目に変更があった場合でも，属性提示部を変更する必要がない．また，属性提示部を新たに実装する場合でも，一般に公開されている XML パーザを流用することができる．

図 29(b) は属性・部分系列受信部が機能実行履歴の部分系列を機能実行履歴サーバに要求する際のデータの流れを表している．履歴を参照するユーザにより，部分系列が要求されると，実行頻度一覧提示部，又は，部分系列一覧提示部を通じて，属性・部分系



(a) 機能実行履歴の属性要求のデータの流れ



(b) 機能実行履歴の部分系列要求のデータの流れ

図 29 機能実行履歴提示のデータの流れ

```

<userList>
  <header>
    <userName type= "string">ユーザ名</userName>
    <numberOfFunction type="number">使用機能の種類</numberOfFunction>
    <commonFunction type="number">共通する機能数</commonFunction>
    <linesOfHistory type="number">履歴の総行数</linesOfHistory>
    <numberOfReceived type="number">受信回数</numberOfReceived>
    <firstArrivedAt type="string">初回受信日時</firstArrivedAt>
  </header>
  <userProperty>
    <user>
      <userName>userA</userName>
      <numberOfFunction>82</numberOfFunction>
      <commonFunction>82</commonFunction>
      <linesOfHistory>523</linesOfHistory>
      <numberOfReceived>16</numberOfReceived>
      <firstArrivedAt>Wed Jan 12 02:37:27 JST 2000</firstArrivedAt>
    </user>
    <user>
      .
      .
    </user>
  </userProperty>
</userList>

```

図 30 機能実行履歴の属性のフォーマット

列受信部が通知を受ける。属性・部分系列受信部は、履歴を参照するユーザが起動しているソフトウェア名、及び、ユーザ名、参照対象ユーザ名を HTTPD 経由で部分系列送出部に送信する。

部分系列送出部は、ファイルシステムに蓄積された各参照対象ユーザの機能実行履歴に含まれる部分系列とその出現頻度をマージし、ユーザ毎の出現頻度を付加した部分系列を、HTTPD 経由で属性・部分系列受信部に送信する。

ユーザ毎の出現頻度を付加した部分系列を受信した属性・部分系列受信部は、実行頻度一覧提示部に長さ 1 の部分系列を、部分系列一覧提示部に長さ 2 以上の部分系列を、それぞれ渡す。実行頻度一覧提示部、及び、部分系列一覧提示部は、渡された部分系列をもとに一覧を作成し、ユーザに提示する。

5. 評価実験

本章では 4章で述べたシステムを利用して利用知識の共有が可能であることを確認することを目的とした評価実験について述べる．評価実験は 2章で述べた同一タスク，類似タスクの両方について実施した．

5.1 評価実験（同一タスク）

5.1.1 概要

CLAS，及び，その支援システムの有効性を確認することを目的として，4章で紹介したシステムを用いた実験を行った．被験者は奈良先端科学技術大学院大学の学生 4 名と秘書 1 名である．なお，3章の被験者とは異なる被験者である．被験者は Microsoft Word 2000 を用いてタスクを実行する．各被験者の Microsoft Word の使用年数，主な作成ドキュメントを表 3 に示す．

被験者に与えられたタスクは，プレーンテキストをある形式の文書に整形することである．タスクの詳細は 5.1.2 で述べるが，タスクの作業効率を高める 5 つの機能がある．これらの機能は被験者には事前に知らせていない．5 つの機能の利用知識がシステムを使用することにより共有されるか評価する．実験終了時に，各人の機能実行履歴を参照しながら，被験者にアンケートとインタビューを行い，タスクを効率よく遂行できる 5 つの機能について，既知であったか，実験中に利用知識を獲得したか，タスク遂行のため利用したか，を確認した．

5.1.2 タスク

被験者に与えるタスクは共通で，Microsoft Word 2000 を用いて（ファイルとして格納されている）プレーンテキストをある形式の文書に整形する作業である．3章の評価実験と形態が似ているが，同一のものではない．被験者には，プレーンテキストのファイル，整形済み文書（できあがり見本）のハードコピーが渡される．整形すべき部分やその具体的な内容は，できあがり見本のハードコピーに指示として書き込まれている．文書のできあがり時のサイズは A4 判 18 ページ，表を含み，日本語で約 16,000 文字，7 章 11 節からなる．

整形作業は次の 4 つの作業（サブタスク）T1 ～ T4 から構成されている．

T1：目次の作成...章，及び，節見出しとページ番号の一覧を作成する．

表 3 同一タスクによる評価実験の被験者の Word 使用経験

被験者	作成ドキュメント	使用年数
S1	手紙, 案内状	1 年
S2	レポート, 案内状	2 年
S3	レポート, 卒論	2 年
S4	レポート, 論文 2	2 年
S5	レポート, 案内状, 論文 1	3 年

T2: 章, 及び, 節見出しのフォント変更 (17カ所) ... 文書全体を通じて, 章と節の見出し部分のフォントを「MS ゴシック」に変更する (本文のフォントは, MS 明朝) .

T3: 表の作成 (3カ所) ... タブで区切られた文字列を指示されたとおり罫線で囲み表を作成する .

T4: 改ページ (14カ所)

Microsoft Word 2000 は, これら 4 つのサブタスク T1 ~ T4 を簡単に遂行することのできる次の 5 つの機能 F1a, F1b, F2 ~ F4 を提供している . F1a は F1b の実行の前にあるあらかじめ実行しておく必要のある機能 (前提機能) である . また, F1a と F2 とを組み合わせること (組合せ機能) により, より短い時間でタスク T2 を実行できる . また, これらの機能の使用によりタスクとして与えた文書の整形で省ける操作の概数を表 4 に示す . F1a は F1b の前提機能, 及び, F2 の組み合わせ機能であるため, F1a 単体では操作数を減らすことはできない . なお, 表 4 中の操作の単位は, 打鍵, 及び, メニュー選択の回数である .

F1a: 「スタイル」... 文書中に目次用の見出しを設定する .

F1b: 「索引と目次」... 文書中の見出しの一覧を, 指定した場所に挿入する . ただし, 機能 F1a で見出しを設定しておく必要がある .

F2: 「スタイル定義」... 文書内の特定の文字列に対して, フォント, フォントサイズ, 行間隔, 文字配置等を一括して変更する . 機能 F1a と併用することにより, 見出し部分のフォント, フォントサイズを再定義できる .

表 4 各機能により省ける操作の概数

機能名	使用時	非使用時
F1a: スタイル	18 操作	—
F1b: 索引と目次	20 操作	1500 打鍵
F2: スタイル定義	1 操作	51 操作
F3: 文字列を表にする	80 操作	300 操作
F4: 改ページ	8 操作	70 打鍵

F3: 「文字列を表にする」...タブなどで区切られた文字列を表に一括変換する。

F4: 「改ページ」...ページを改める。

なお、被験者は、これらサブタスク T1 ~ T4 と機能 F1a ~ F4 の存在を知らされていない。

5.1.3 実験手順

被験者にタスク、支援システムの使用方法を説明する。なお、被験者に対しては、次のような指示が出される。

- Microsoft Word 2000 の既知の機能をできるだけ利用し、効率よくタスクを実行するように。
- 必要があればオンラインヘルプを参照しても構わない。
- 実行が困難と思われる作業は後回しにしても構わない。

実験は次の三つのフェーズで構成されている。

フェーズ 1 (p1): タスク遂行

約 20 分間個別にタスクを実行してもらい、機能実行履歴を収集する。






フェーズ 2 (p2): 機能実行履歴参照

被験者に、フェーズ 1 で収集した機能実行履歴を支援システムを用いて参照してもらい、タスク遂行に役に立ちそうな機能の利用知識の獲得を試みてもらう。参照時間の制約は特に設けない。

フェーズ 3 (p3): タスク遂行, 必要に応じて, 機能実行履歴参照

被験者にタスクを再開してもらい、その終了まで作業を続けてもらう。なお、必要であれば、機能実行履歴を引き続き参照することができる。

5.1.4 結果

4 つのサブタスクの遂行に役立つ 5 つの機能について、各被験者がどのフェーズでその利用知識を獲得し、タスク遂行に利用したかを図 31 に示す。図 31 において、印は利用知識の獲得を、印は機能の実行を、それぞれ表す。印のみの機能は、被験者にとってその利用知識が既知であったことを表す。また、印も 印もない機能は、利用知識の獲得もタスク遂行のための実行もなされなかったことを表す。

サブタスク遂行に役立つ機能はのべ 25 (5 × 5) 個で、そのうち、被験者にとって利用知識が既知であった機能はのべ 9 個、実験中に利用知識が獲得されなかった機能はのべ 6 個であった。実験中に利用知識が獲得された機能はのべ 12 個で、そのうち、他の被験者の機能実行履歴を参照することで利用知識が獲得された機能はのべ 11 個、オンラインヘルプで利用知識が獲得された機能は 1 個のみであった。なお、試作システムを用いて他の被験者の機能実行履歴を参照することで利用知識が獲得された機能のうちの 2 つは、サブタスク遂行に役立つと予め想定した機能とは別の機能であった。

表 2 と図 31 の情報、更に、実験後に被験者に対して行ったアンケートとインタビューの結果に基づいて、個々の被験者の特徴を簡単にまとめると次のようになる。

被験者 S1: 5 つの機能 F1a ~ F4 いずれの利用知識も持たなかった。しかし、他の被験者の機能実行履歴を参照することで、F3, F4 の利用知識を獲得し、フェーズ 3 においてサブタスク T3, T4 遂行のために実行している。機能 F1b の存在には気づいたが、機能 F1a がその前提機能であることに気づかなかったため、機能 F1b を実行することはできなかった。機能 F2 の利用知識の獲得はできなかった。

被験者 S2: 3 つの機能 F1a, F1b, F4 の利用知識を持っていた。また、F1a が F1b の前提機能であることを知っていた。機能 F2 についてはその存在は想像していたが、他の被験者の機能実行履歴の部分系列を参照することで、機能 F2 が F1a の組合せ

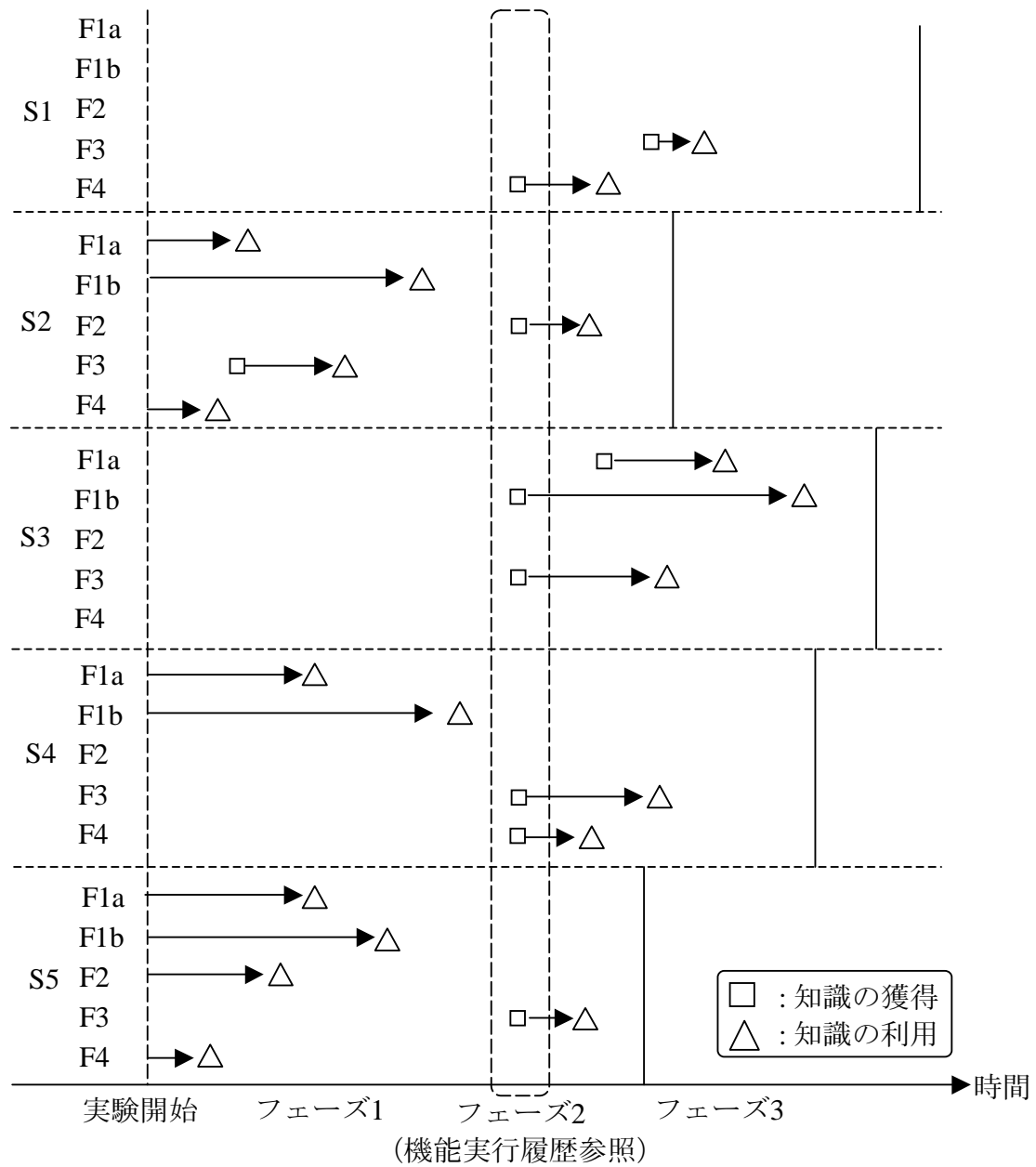


図 31 各被験者の利用知識の獲得時期

機能であることを知り、サブタスク T2 のために使用している。機能 F3 については、フェーズ 1 においてオンラインヘルプを参照することで、その利用知識を獲得し、サブタスク T3 遂行のために実行している。5 つの機能 F1a ~ F4 を全て利用してサブタスクを実施した被験者の一人である。

被験者 S3: 被験者 S1 と同様に F1a ~ F4 いずれの機能の利用知識も持っていなかった。機能毎の実行頻度一覧により、フェーズ 2 において F1b の存在に気づき、フェーズ 3 において部分系列の一覧から F1a が F1b の前提機能であることを知り、サブタスク T1 のために使用している。機能 F2, F4 の利用知識は獲得できなかった。

被験者 S4: F1a と F1b の機能の存在、F1a が F1b の前提機能であることを知っており、フェーズ 1 でサブタスク T1 を遂行するために実行している。フェーズ 2 において、機能毎の実行頻度一覧から機能 F3, F4 の存在に気づき、フェーズ 3 においてサブタスク T3, T4 のために実行している。

被験者 S5: 機能 F3 以外の機能の存在、F1a が F1b の前提機能であること、及び、F1a が F2 の組合せ機能であることを知っていた。フェーズ 2 において単一機能の一覧から機能 F3 の存在を知り、フェーズ 3 においてサブタスク T3 のために実行している。5 つの機能 F1a ~ F4 を全て利用してサブタスクを実施した被験者の一人である。

5.1.5 評価実験（同一タスク）の考察

CLAS、及び、支援システムがソフトウェア利用知識の共有に役立つ可能性のあることを示唆する次のような結果が得られた。

- 機能毎の実行頻度一覧は初心者（被験者 S1）を含め、すべての被験者に対して存在に気づいていない機能の発見を支援した。また、機能毎の実行頻度一覧を閲覧する時間は 10 分程度であり、比較的短い時間で利用知識を獲得することができた。機能毎の実行頻度一覧は、小さいコストで初心者から熟練者までの利用知識の共有に寄与する可能性がある。
- 4 人の被験者が、ある機能の前提機能や組合せ機能を知らなかった。このうち、2 人の被験者が部分系列の一覧を参照することにより、既知の機能の前提機能、組合せ機能の存在に気づき、実行することができた。部分系列の一覧により、存在に

表 5 類似タスクによる評価実験の被験者の Word 使用経験

被験者	使用年数	作成ドキュメント
S1	3 年	論文
S2	4 年	論文, ソフトウェア設計書
S3	2 年	論文, ソフトウェア設計書
S4	2 年	論文, 卒論, レポート
S5	2 年	論文, 卒論, レポート

は気づいているが、前提機能を知らないため実行できない機能、及び、組み合わせることで作業効率を高めることができる機能の存在を知ることができる場合がある。

- 機能実行履歴には、実行されたメニュー項目名だけでなく、メニュー階層内での位置が分かるよう、ルートメニューから実行されたメニュー項目名までの選択経路が付加されている。こうした情報は、ユーザがオンラインヘルプ等で機能の詳細説明を検索する際の非常に適切なキーワードとなる。もちろん、ユーザがその機能を試用しようと思えば、示された選択経路に従って実際にメニュー項目を選択するだけでよい。オンラインヘルプ等の既存のユーザ支援機構と提案方式をうまく連携させることで、利用知識の獲得コストを大幅に小さくできる可能性がある。

5.2 評価実験（類似タスク）

5.2.1 概要

対象ソフトウェアを、Microsoft Word 2000 と PowerPoint 2000 とし、支援システムを用いた実験を行った。被験者は、普段、これらのソフトウェアを論文執筆、研究発表用スライド作成に使用している。実験のあらまは以下のとおりである。

- 被験者は奈良先端科学技術大学院大学情報科学研究科教官 1 名、学生 5 名の計 6 名である。
- 被験者の以前のバージョンを含めた Word, PowerPoint の使用年数、及び、主な作成ドキュメントを表 5, 6 に示す。

表 6 類似タスクによる評価実験の被験者の PowerPoint 使用経験

被験者	使用年数	作成ドキュメント
S1	4 年	輪講，研究発表用スライド，研究発表用ポスター
S2	4 年	輪講，研究発表用スライド
S3	3 年	輪講，研究発表用スライド，研究発表用ポスター
S6	4 年	輪講，研究発表用スライド

- 被験者からは，あらかじめ 1 ～ 10 カ月間にわたり，論文執筆，研究発表用スライド作成時の機能実行履歴を収集している．
- 各被験者に提案システムを用いて次節で述べるタスクを実行してもらい，未知の機能の発見を試みてもらった．また，発見した機能が自分にとって有用であるか（今後，論文作成やスライド作成において使用するか）どうかについて答えてもらった．

5.2.2 タスク

システムは，Word と PowerPoint のそれぞれについて，図 21 に示すような全被験者の機能毎の実行頻度一覧をユーザに提示する．被験者は，システムより提示された機能のうち，自分の実行頻度がゼロとなっている機能を選んで，それらの機能の中から未知機能の発見を試みる．つまり，他の被験者のいずれかが 1 度以上実行した機能のうち自分が 1 度も実行したことの無い機能がシステムより提示される．

各被験者には，まず，システムが提示した各機能に対し，既知であったか，未知であったかを答えてもらう．次に，未知であった場合には，その機能を含む部分系列の一覧（図 27）の参照，機能の実行，オンラインヘルプの参照，のいずれかにより，機能のはたらきを理解してもらう．機能のはたらきを理解した上で，その機能が有用か（今後，論文執筆やスライド，ポスター作成の際に，その機能を使おうと思うか）を答えてもらう．

実験で使用した Word 2000，PowerPoint 2000 の機能実行履歴の概要を，それぞれ表 7，表 8 に示す．Word 2000 の機能実行履歴は，論文を執筆する目的で使用したものだけが含まれる．PowerPoint 2000 の機能実行履歴は，輪講，研究発表用スライド，及び，ポスターを作成する目的で使用したものだけが含まれる．表 7，8 に示されるように，履歴の行数が多いユーザ（すなわち，より多くの機能を実行したユーザ）ほど実行機能の

表 7 実験に使用した Word 2000 の機能実行履歴

被験者	蓄積期間	履歴の行数	機能数
S1	1 カ月	576	72
S2	10 カ月	2318	59
S3	9 カ月	657	92
S4	2 カ月	268	28
S5	2 カ月	298	43
全被験者	—	4117	123

表 8 実験に使用した PowerPoint 2000 の機能実行履歴

被験者	蓄積期間	履歴行数	機能数
S1	1 カ月	671	86
S2	1 カ月	1114	69
S3	2 カ月	1231	91
S6	1 カ月	281	22
全被験者	—	3297	130

種類数が多いとは限らない。また、各ユーザの実行機能の種類数には最大約 4 倍の差があったが、他のユーザが実行した機能を全て包含して実行したユーザはいなかった。なお、Word における各ユーザの実行機能の種類は 123 種類であり、Powerpoint では 130 種類であった。

5.2.3 結果

実験結果を表 9、及び、表 10 に示す。表 9、10 中の「提示機能数」は、システムが提示した機能の数であり、ある被験者が実行したことのない機能のうち、他の被験者のいずれかが 1 度以上実行した機能の数である。「既知の機能数」は、提示機能数のうち被験者が既に利用知識を獲得していたと答えた機能の数である。「獲得機能数」は、被験者が実験中に利用知識を獲得した機能の数である。「未獲得の機能数」は、利用知識が獲得でき

表 9 実験結果 (Word)

被験者	提示機能数	既知の機能数	獲得機能数	未獲得の機能数	有用と判断した機能数
S1	51	15	28	8	9 (32%)
S2	64	48	9	7	5 (56%)
S3	31	0	18	13	5 (28%)
S4	95	69	23	3	9 (39%)
S5	80	50	24	6	12 (50%)
平均	64.2	36.4	20.4	7.4	8 (39.2%)

なかった機能であり，機能の実行方法がわからず，部分系列を見ることによっても，コンテキストを理解できず，オンラインヘルプによっても説明を得られなかった機能である。「有用と判断した機能」は，利用知識を獲得した機能（獲得機能）のうち，論文執筆，研究発表用スライド，ポスター作成において有用であり，今後使用する可能性が高いと被験者が答えた機能の数である。「有用と判断した機能」のカッコ内の割合は，「獲得機能数」に対する「有用であると判断した機能数」の割合である。

表 9，10 に示されるように，実験に使用した両ソフトウェアにおいて，被験者全員が機能の利用知識を獲得でき，そのうちのいくつかについては今後使用するつもりであると答えた。Word では，被験者が利用知識を獲得した機能のうち有用と判断した機能が平均で 39.2%であった。PowerPoint では，被験者が利用知識を獲得した機能のうち有用と判断した機能は平均で 46.3%であった。

実験後の被験者から以下のような感想が得られた。

- システムにより提示された機能名だけでは，機能のはたらきが想像できないものであっても，その機能を含む部分系列を見ることにより，機能の働きを理解できるものがあつた。
- システムが提示する機能には，contextual メニューなど状況に依存して現れるメニュー項目があり，それらは機能名とその選択経路だけでは機能実行を再現できないものがあつた。そのような機能であっても，機能名をキーワードとしてオンラインヘルプを検索することで実行方法を知ることができる場合があつた。

表 10 実験結果 (PowerPoint)

被験者	提示機能数	既知の機能数	獲得機能数	未獲得の機能数	有用と判断した機能数
S1	44	24	16	4	4 (25%)
S2	61	36	9	16	6 (67%)
S3	39	33	3	3	2 (67%)
S6	108	76	15	17	8 (53%)
平均	63.0	42.3	10.8	10.0	5.0 (46.3%)

- システムにより提示された機能名をキーワードとしてオンラインヘルプを検索しても、検索結果にその機能の説明が含まれず、機能のはたらきが理解できない場合があった。

5.2.4 評価実験（類似タスク）の考察

評価実験により、CLAS、及び、支援システムが利用知識の獲得を支援するために有効であることを示唆する結果が得られた。システムが被験者に提示した機能数の平均は63.4であった。この数はWordやPowerPointが提供している機能の総数（Wordは約500個、PowerPointは約420個）と比べると少ない。支援システムを用いることで、ソフトウェアの提供する膨大な数の機能の中から各ユーザにとって有用な機能を選び出せることが示唆された。なお、システムが提示した機能の中には被験者が既知であった機能も含まれていた。これは、機能実行履歴の収集期間が短かったため、既知であるにも関わらず実行頻度がゼロとなっていた機能が存在したためである。各被験者が継続して機能実行履歴を収集することで、システムが提示する機能数を小さくすることが期待できる。

被験者が利用知識を獲得した機能の約半数は、被験者が今後使うであろうと回答した（有用な）機能であった。被験者全員がその存在にすら気づいていなかった有用な機能を発見することができた。

ある機能を実行するコンテキストを理解するためには、部分系列の一覧が役立った。評価実験においては、機能名をキーワードとしてオンラインヘルプを検索しても、その機能の説明が得られない機能がいくつか存在したが、その機能を含む部分系列において共起、連鎖する機能の機能名から、その機能がどのようなものが想像することができる場

合があった。例えば、機能名「レベル上げ」だけでは、その機能がどのようなものか判断できなかったが、その前後に実行された機能の機能名「インデント」を見ることにより、「レベル上げ」がインデントの深さを変更する機能であることがわかった。

現状の問題点は、表9, 10に示されるように、「未獲得の機能」が存在したことである。つまり、各ユーザはシステムが提示した機能全ての利用知識を獲得できたわけではなかった。システムは機能名をユーザに提示したが、機能名やその系列からだけでは（オンラインヘルプを併用しても）その機能のはたらきが理解できず、利用知識の獲得に至らなかった場合があった。そのような機能の多くは、操作方法がわからず、機能実行を再現できない機能であった。例えば、状況依存で現れるメニュー、特に contextual メニューはその機能の実行を再現するのが難しく、その結果、機能の発見に至らなかった。contextual メニューは機能実行の対象（表、テキスト、等）に依存して現れる場合が多い。機能実行履歴に機能実行対象の情報を付加するなどして、各機能の実行を再現するためのシステムを構築することは今後の課題の一つである。

6. 関連研究

6.1 知識共有

ソフトウェアの利用知識を対象としたものではないが，個人のノウハウや知識を組織で共有することを目指したシステムは多い [28][38][39]．個人の暗黙知を有形化し，グループで共有するシステムとしては Advice/Help on Demand が知られている [28]．業務ノウハウを文書化しグループ内での参照を可能にしている．ただし，業務ノウハウが自然言語で記述されているため，CLAS のように，ユーザの本来の作業を妨げずに，収集，共有することは容易ではない．

似通った興味を持つコミュニティ内で知識を共有する方法として協調的情報フィルタリング [12][27] がある．協調的情報フィルタリングでは，似通った興味を持つユーザ間で協力し合いながら自分の得た情報を評価することにより，自分，及び，他ユーザの情報をフィルタリングすることを手助けする．

協調的情報フィルタリングを web に応用した研究は多い [7][21][22][36]．例えば，Siteseer [36] やブックマークエージェント [23] はブックマークがユーザの興味のある web ページを表しているとして，興味が似通ったユーザ間で URL を共有する．ブックマーク中に共通する URL からユーザの類似度を判定し，類似したユーザ同士でお互いに知らない URL を知らせる．各 URL の類似度は URL が示している HTML ファイルから抽出されるキーワードにより求める．ShopBot [7] は協調的情報フィルタリングを e-commerce サイトの商品情報に応用している．ただし，ユーザの興味が機能実行履歴として自動収集可能な CLAS と異なり，web ページに対する興味をユーザがシステムに明示的に知らせる必要がある．

6.2 操作履歴

本提案手法と同様に，ソフトウェアユーザの行動やソフトウェアを利用した業務内容を記述する手段として，ソフトウェアの実行（操作）履歴を用いる研究は数多く報告されている [2][8][19][24][29]．例えば，森らは，X-window におけるマウス操作や打鍵などの詳細な操作履歴を収集，分析するツールを提案している [24]．彼らのシステムを用いれば，操作履歴に基づいて，ユーザ操作の再現，正規表現に似た形式での操作履歴の分析が可能である．また，安部田らは，個人情報管理システム（PIM）の操作履歴を収集し，企業における業務知識を体系化するシステムを提案している [1]．

ソフトウェアの実行（操作）履歴をネットワーク経由で収集するシステムはいくつか提案されている。Hilbertらは、ソフトウェア設計者の意図する操作系列と異なる操作をユーザが行うと、その詳細を電子メールで設計者に通知するシステムを開発している [14]。彼らのシステムを用いれば、ソフトウェア操作に対する設計者とユーザの認識のずれを知ることができ、ソフトウェア設計に役立てることができる。但し、ユーザの操作と比較するための操作系列を設計者が予めシステムに登録しておく必要がある。CLASのように、実際の機能実行履歴の中から利用知識を抽出することはできず、ユーザ間での利用知識の共有を支援する機能も有していない。

6.3 ユーザ支援

DYK(Did You Know) システム [33] や Tip of the Day [15] は、ソフトウェアが提供する機能をユーザに知らせることを目的としている。具体的には、ソフトウェアが提供する機能の一つを選択し、その使い方をソフトウェア起動時などにユーザに提示する。選択された機能がユーザにとって未知であれば、新たな機能を知らせることができるが、提示する機能がユーザにとって既知であるかどうかの判断基準については、特に言及されていない。

ユーザの操作履歴からユーザの置かれた状況を推測し、状況に応じてシステムがユーザを支援するシステムの研究は多く行われているが、いずれも操作履歴からユーザの置かれた状況を推測するためのユーザモデル [9] の構築が必要になる。文献 [13] や [17] で提案されている方式では、自然言語で入力されたユーザの質問に操作履歴から推測されるユーザの置かれた状況を推測し、推測された状況に適したヘルプを提示する。

文献 [10], [30], [31], [46], [48], 及び, [49] のように、ユーザの操作の度にその操作、それまでの操作履歴、及び、ユーザモデルからユーザの置かれた状況を推測し、ユーザがソフトウェアが提供する機能に気づいていないと判断された場合に、積極的にその機能をユーザに知らせるシステムも提案されている。しかし、いずれもユーザの意図を推測するためのユーザモデルの構築が必要である。特に、CLASが対象とする機能が肥大化したソフトウェアでは、ユーザモデルの構築は容易ではない。

予測インタフェース [6][20][37] や例示インタフェース [16][43] は、ユーザの作業効率の向上を目的として、操作履歴に含まれる繰り返しや一定のパターンを発見し、次の操作を予測したり、一連の操作を一括してマクロ定義する。ただし、予測やマクロ定義は、収集したユーザ個人のために使用され、CLASのように他ユーザに使われることを想定し

ているわけではない。

7. 結論

本論文では、幅広いユーザ層を想定した多機能アプリケーションソフトウェアが提供する機能の利用知識獲得コストを小さくすることを目的として、ユーザ間で利用知識を共有する方式 *CLAS* を提案した。*CLAS* は、ソフトウェア機能の利用知識の獲得を支援する。ここで、ソフトウェア機能の利用知識は、ソフトウェアが提供する機能の存在に気づき、機能実行のための操作方法を理解し、その機能がどのような状況で有効に働くかを理解することである。

利用知識共有は 3 つの手順から構成される。まず、似通った目的で同一ソフトウェアを使用するユーザでコミュニティを形成し、コミュニティに属するユーザのソフトウェア機能の実行履歴（コマンド実行履歴）を収集する。収集の際に行なうタスクは、同一タスク、または、類似タスクのいずれかである。次に、蓄積した機能実行履歴を相互参照する。相互参照する機能を、履歴を参照するユーザが実行したことの無い機能のうち、他ユーザの多くが実行したことがある機能、及び、他ユーザのいずれかの実行頻度が大きい機能とすることにより、履歴を参照するユーザにとって未知であった有用な機能を発見することができる。特に同一タスクを実行した際に収集した機能実行履歴を参照する際には、履歴を参照するユーザしか実行していない機能を参照し、その機能の代替となる機能を発見できる。最後に、発見した機能名（コマンド名）を手がかりとし、機能実行履歴の部分系列やオンラインマニュアルを検索するなどして、発見した機能の利用知識を獲得することができる。

CLAS に基づく試作システムを用いた評価実験において利用知識が共有できることを確認した。評価実験では、Microsoft Word 2000 を対象ソフトウェアとし、被験者 5 名のうち 4 名において、のべ 13 個の未知機能からのべ 8 個の機能の利用知識の獲得が認められた。初心者、熟練者に関わらず、その存在を想像さえしていなかった機能について利用知識の獲得がなされた。また、試作システムに関して、機能実行履歴提示部の利便性、提示する情報、プライバシーの 3 つの問題点が明らかになった。

試作システムを用いた評価実験で得られた知見に基づき、利用知識共有支援システムを開発した。利用知識共有支援システムにより、蓄積量の大きい機能実行履歴からもユーザが利用知識の獲得ができることを目指した。特に、試作システムを用いた評価実験で問題となった膨大な量の機能実行履歴の部分系列を、単一機能（長さ 1 の部分系列）の実行頻度、履歴を参照するユーザが指定した機能を含む部分系列のみを提示することにより、重要な部分を失うことなく大幅に減少できた。

利用知識共有支援システムを用いた評価実験では、同一タスク、及び、類似タスクにおける利用知識の共有を確認した。同一タスクにおける評価実験では、5名の被験者全員がのべ19個の未知機能のうち11個の機能の利用知識が獲得された。同一タスクにおける評価実験では、被験者に知らされていないサブタスク、及び、サブタスクを効率的に遂行できる機能を設定した。サブタスクの間には、あらかじめ実行しておく必要のある機能（前提機能）、組合せると作業効率が高まる機能（組合せ機能）があった。支援システムにより、被験者にとって未知であった前提機能、及び、組合せ機能を獲得することができた。類似タスクにおける評価実験では、2種類のソフトウェアに対し1～10ヶ月に渡り蓄積した機能実行履歴から、のべ9名の被験者が60個の機能の利用知識を獲得した。また、獲得した60個の機能のうち、被験者が有用と判断した機能は、約半数であることが示された。

CLASでは、似通った目的を持ち同一ソフトウェアを使用するユーザ間で機能実行履歴を収集し、ユーザは収集した機能実行履歴を相互に参照する。機能実行履歴の収集によるユーザへの負担は極めて小さい。また、ユーザの指定により機能実行履歴を随時参照できる。利用知識共有支援システムが提示する未知の有用な機能の候補のすべてが有用であるとは限らないが、候補の一覧を参照することによりユーザは未知の有用な機能を選び出すことができる。そのため、active help systemなどのユーザ支援システムのように膨大な数のソフトウェア機能に対して複雑で推測精度の高いユーザモデルを構築する必要がない。

利用知識共有支援システムは、既存のヘルプシステムなどの他のユーザ支援システムと競合するものではない。つまり、従来のヘルプシステムを使うか提案システムを使うかの二者択一をユーザに迫るわけではない。支援システムはユーザにとって有用と思われる機能の発見には役立つが、発見した機能の使い方の詳細を調べるためには、既存のヘルプシステムを併用することがむしろ望まれる。

支援システムを用いた機能実行履歴の参照はユーザの明示的な指示のもとで行われるが、システム側から積極的に提示することも可能である。提示のタイミングを含めた機能実行履歴の提示方式を考案することは今後の課題の1つである。また、機能毎の実行頻度一覧では、各機能に対してユーザ毎の実行頻度を提示したが、これら全てが必ずしも必要な情報であるとは限らない。提案方式では、短時間で未知の機能を知ることが目的としているため、ユーザにとって有用な情報を残しながら、提示量を減らすことも今後の課題の1つである。

現状では GUI を持つソフトウェアのメニュー選択、及び、同様の結果を得られる操作を機能実行履歴として収集している。しかし、機能実行履歴の収集はメニュー選択に限定されるものではない。異なる種類の機能実行履歴の収集方式、及び、その提示方式を用いることにより、提案方式の適用範囲を大きくすることが可能である。

参考文献

- [1] 安部田 章: ユーザの操作履歴から業務知識を抽出する個人情報管理システム, 情報処理学会 研究報告, GW23-2, pp.7-12, 1997.
- [2] 赤池 英夫, 角田 博保: X-Window 上の利用者行動分析システム, 情報処理学会論文誌, Vol. 33, No. 5, pp. 736-745, 1992.
- [3] Belkin J. Nicholas: Helping People Find What They Don't Know, Communications of the ACM, Vol. 43, No. 8, pp.58-61, 2000.
- [4] J. M. Carroll and M. B. Rosson : Paradox of the Active User, Interfacing Thought: Cognitive Aspects of Human-computer Interaction, MIT Press, 1987.
- [5] R. Chimera: Value Bars: An Information Visualization and Navigation Tool for Multi-attribute Listings, Proceedings of the ACM Conference on Human Factors in Computing Systems(CHI'92), pp.293-294, 1992.
- [6] J. J. Darragh, I. H. Witten, and M. L. James: The Reactive Keyboard: A Predictive Typing Aid, IEEE Computer, Vol. 23, No. 11, pp. 41-49, 1999.
- [7] R. B. Doorenbos, O. Etizoni, and D. S. Weld: A Scalable Comparison-shopping Agent for the World Wide Web, Proceedings of 1st International Conference on Autonomous Agents, pp. 39-48, 1997.
- [8] 海老名 毅, 伊藤 昭: X アプリケーションにおける可読性のある操作履歴の取得について, 電子情報通信学会 技術研究報告, HC94-87, pp. 1-7, 1995.
- [9] G. Fischer: User Modeling: The Long and Winding Road, Proceedings of User Modeling Conference 99, pp.349-355, 1999.
- [10] G. Fischer, A. C. Lemke, and T. Schwab: Knowledge-based Help Systems, Proceedings of Human Factors in Computing Systems 85, pp.161-167, 1985.
- [11] G. Fischer, 杉本 雅則(訳): グループには頭がない: ソーシャルインタラクションを支援するための概念的枠組みとシステム, 情報処理, Vol. 40, No. 6, pp. 575-582, 1999.

- [12] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry: Using Collaborative Filtering to Weave an Information TAPESTRY, *Communications of the ACM*, pp.61–70, Dec. 1992.
- [13] M. Hecking: How to Use Plan Recognition to Improve the Abilities of the Intelligent Help System SINIX Consultant, *Proceedings of INTERACT*, pp. 657–662, 1987.
- [14] D. M. Hilbert and D. F. Redmiles: An Approach to Large-scale Collection of Application Usage Data Over the Internet, *Proceedings of 20th International Conference on Software Engineering*, pp.136–145, 1998.
- [15] E. Horvitz: Agents with Beliefs: Reflections on Bayesian Methods for User Modeling, *Proceedings of User Modeling Conference 97*, pp.441–442, 1997.
- [16] S. E. Hudson and C. N. Hsi: A Synergistic Approach to Specifying Simple Number Independent Layouts by Example, *Proceedings of Conference on Human Factors in Computing Systems*, pp. 285–292, 1993.
- [17] 伊藤 昭, 海老名 毅, 熊本 忠彦: 対話型計算機利用支援におけるユーザ質問の分類と支援回答戦略, *電子情報通信学会論文誌*, Vol. J77-D-II, No. 7 pp.1319–1328, 1994
- [18] T. Joachims, D. Freitag, and T. Mitchell: Webwatcher: A Tour Guide for the World Wide Web, *Proceedings of International Joint Conference on Artificial Intelligence*, Vol. 1, pp. 770–775, 1997.
- [19] 前田 恵三, 中野 靖夫: コンピュータ操作過程の記録・分析支援ツール, *情報処理学会 技術報告*, HI-81-2, Vol. 98, No. 116, 1998.
- [20] T. Masui and K. Nakayama : Repeat and Predict — Two Keys to Efficient Text Editing, *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Addison-Wesley, pp. 118–123, 1994.
- [21] D. M. Maurice , S. A. Sarabjot, and G. B. Alex: Personalization on the Net Using Web Mining, *Communications of the ACM*, Vol. 43, No. 8, pp.123–125, 2000.

- [22] B. Mobasher, R. Cooley, and J. Srivastava: Automatic Personalization Based on Web Usage Mining, *Communications of the ACM*, Vol. 43, No. 8, pp. 142–151, 2000.
- [23] 森 幹彦, 山田 誠二: ブックマークエージェントの共有による情報検索の支援, *電子情報通信学会論文誌 D-1*, Vol.83, No. 5, pp.487–494, 2000.
- [24] 森 孝弘, 西田 知博, 斎藤 明紀, 都倉 信樹: 大量の GUI 操作履歴を分析するための走査・再生ツール, *情報処理学会研究報告*, HI-69-1, pp. 1–8, 1996.
- [25] 森崎 修司, 門田 暁人, 松本 健一, 井上 克郎, 鳥居 宏次: 機能実行履歴を用いたソフトウェア利用知識の共有, *情報処理学会論文誌*, Vol.41, No. 10, pp. 2770–2781, 2000.
- [26] S. Morisaki, A. Monden, K. Matsumoto, K. Inoue, and K. Torii: Two-level User Interface Evolution by Sharing Usage Logs, *Proceedings of International Workshop on Software Evolution*, pp. 70–73, 1999.
- [27] 森田 昌宏, 速水 治夫: 情報フィルタリングシステム, *情報処理*, Vol. 37, No. 8, pp.751–757, 1996.
- [28] 中山 康子, 真鍋 俊彦, 竹林 洋一: 知識情報共有システム (Advice/Help on Demand) の開発と実践:知識ベースとノウハウベースの構築, *情報処理学会論文誌*, Vol.39, No.5, pp. 1186–1191, 1998.
- [29] A. Neal and R. Simons: Playback: A Method for Evaluating the Usability of Software and Its Documentation, *Proceedings of Human Factors in Computing Systems 83*, pp.78–82, 1983.
- [30] 西村 俊和, 広瀬 勝一, 美濃 導彦, 池田 克夫: マン・マシンコミュニケーションにおける文脈の追跡, *電子情報通信学会論文誌*, Vol. J78-D-II, No. 1, pp. 105–114, 1995.
- [31] 西村 俊和, 美濃 導彦, 池田 克夫: コマンド予測シェル使用時のユーザのキー入力の負担, *情報処理学会研究報告*, HI-69-5, pp. 33–40, 1995.

- [32] 大山 隆一郎, 山本 喜一: テキストエディタにおける能動的ヘルプシステム, コンピュータソフトウェア, Vol. 13, No. 5, pp.11-26, 1996.
- [33] D. Owen: Answers First, Then Questions, User-Centered System Design, New Perspectives on Human-Computer Interaction, Lawrence Erlbaum Associates, Inc., pp.361-375, 1986.
- [34] R. Rao and S. K. Card. : The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information, Human Factors in Computing Systems (CHI'94), pp.318-322, ACM Press, 1994.
- [35] D. Reisberg: Cognition , W. W. Norton & Company, 1997.
- [36] J. Rucker and M. J. Polanco: Siteseer: Personalized Navigation for the Web, Communications of the ACM, Vol. 40, No. 3, pp.73-75, 1997.
- [37] A. Sears and B. Shneiderman: Split Menus: Effectively Using Selection Frequency to Organize Menus, ACM Transaction on Computer-Human Interaction, Vol. 1, No. 1, pp. 27-51, 1994.
- [38] 関 良明, 山上 俊彦, 清水 明宏: ノウハウ蓄積支援システム FISH の実現とその評価, 電子情報通信学会 論文誌 , Vol. J76-D-II , No. 6, pp. 1223-1232, 1993.
- [39] 敷田 幹文, 門脇 千恵, 國藤 進: フローに連携した組織内インフォーマル情報共有手法の提案, 情報処理学会論文誌 , Vol. 41, No. 10, pp. 2731-2741, 2000.
- [40] 白石 裕美, 森崎 修司, 門田 暁人, 井上 克郎: 未知機能の発見支援を目的としたソフトウェア機能実行履歴の提示, 情報処理学会研究報告 , No. 97, 2000-GW-37-3, pp.13-18, 2000.
- [41] J. L. Steffen, S. G. Eick, E. E. Sumner Jr: Seesoft - A Tool for Visualizing Line-oriented Software Statistics, IEEE Transaction on Software Engineering, Vol. 18, No. 11, pp.957-968, 1992.
- [42] G. E. Stephen, C. N. Michael, and D. S. Jeffery: Graphical Analysis of Computer Log Files, Communications of the ACM, pp. 50-56, Dec. 1994.

- [43] 杉浦 淳, 古関 義幸: 例示プログラミングにおけるマクロ定義の簡略化, 日本ソフトウェア科学会 インタラクティブシステムとソフトウェア IV, 近代科学社, pp. 101–110, 1996.
- [44] 高田 光男, 小高 知宏, 小倉 久和: UNIX シェルユーザのための適応型ユーザインタフェースの構築とその評価, 第 50 回情報処理学会全国体会論文集 (4), pp.331–332, 1995.
- [45] 高田 哲司, 小池 英樹: ログ情報視覚化システムを用いた集団監視による不正侵入対策手法の提案, 情報処理学会 論文誌, Vol. 41, No. 8, pp.2216–2227, 2000.
- [46] 高田 光男, 西野 順二, 小高 知宏, 小倉 久和: UNIX 高機能シェルの行編集機能に対する適応型ヒューマンインタフェースの構築とその評価, 情報処理学会 論文誌, Vol. 38, No. 10, pp.1919–1927, 1997.
- [47] K. Torii, K. Matsumoto, K. Nakakoji, Y. Takada, S. Takada and K. Shima: Ginger2: An environment for CAESE (Computer-Aided Empirical Software Engineering), IEEE Transanction on Software Engineering, Vol.25, No. 4, pp.474–492, 1999.
- [48] R. Wilensky, Y. Arens, and D. Chin: Talking to UNIX in English: An Overview of UC, Communications of ACM, Vol. 27, No. 6, pp.574–593, 1984.
- [49] R. Winkels, J. Breuker, and N. denHaan: Principles and Practice of Knowledge Representation in EUROHELP, Proceedings of International Conference on the Learning Sciences, pp.442–448, 1991.
- [50] A. Y. Zissos, and I. H. Witten: User Modeling for a Computer Coach: a Case Study, International Journal of Man-machine Study, Vol. 23, pp. 729–750, 1985.

謝辞

本研究を進めるにあたり，多くの方々の有用なご指導，ご支援を頂きました．ここに謝意を表します．

奈良先端科学技術大学院大学 情報科学研究科 井上 克郎 教授には博士後期課程 2，3 年次において，常に簡潔，かつ，建設的なご意見，論文構成上のご助言をはじめとして多大なご指導，ご助力を頂きました．なかでも評価実験に関するご指摘はたいへん有益であり，本研究において，観点の異なる 2 つの評価実験を実施することができました．また，博士前期課程，博士後期課程 1 年次においても多くのご助言，ご配慮，興味深いお話しを頂きました．心より感謝致します．

本論文の執筆にあたり，貴重なご指導，ご助言を頂きました奈良先端科学技術大学院大学 情報科学センター 小山 正樹 教授，奈良先端科学技術大学院大学 情報科学研究科 関 浩之 教授に心より感謝致します．小山 正樹 教授には，本研究のビジネスとしての可能性，及び，本研究が文化や民族に関わらず適用可能であるかについてご説示頂きました．関 浩之 教授には細部に及ぶ鋭いご指摘を頂きました．特に，数式による表現に対するご指摘はたいへん有益でした．また，頂いたご質問，ご助言は，本研究の新たな方向性を示唆するものであり，今後の本研究の発展に大きく寄与するものと考えます．

奈良先端科学技術大学院大学 情報科学研究科 松本 健一 助教授には本研究の全過程において多大なご指導，ご助力を，連休，休，平日，昼夜を問わず，頂きました．研究者としての姿勢を研究を通じてご教示頂いたこと，心強いご助言を頂いたこと，博士後期課程への進学に関して励ましのお言葉を頂いたこと，及び，締切りの度にご心配，ご迷惑をおかけしたにもかかわらず，我慢強く接して頂いたことに深く感謝致します．また，研究だけでなく生活面においても多大なご配慮，ご支援を頂いたことにも深く感謝しております．

奈良先端科学技術大学院大学 鳥居 宏次 副学長には，博士前期課程，及び，後期課程 1 年次において，多くのご指導，ご助言，叱咤激励を頂きました．頂いた数々のご助言は常に研究の発展を見据えたものであり，本研究の発展にも必要不可欠なものでした．また，博士後期課程進学に際して，貴重なご助言を頂きました．博士後期課程 2，3 年次においても，就業時間外にお時間を頂き，研究に関する相談に応じて頂いたこと，国際会議参加において貴重な経験をする機会を設けて頂いたこと，及び，生活面においても多大なご配慮頂いたことに深く感謝致します．

奈良先端科学技術大学院大学 情報科学センター 飯田 元 助教授には，博士前期課程に

において本研究の礎となる研究を進める上で多大なご指導，ご助言を頂きました．研究の進め方に関するご指導，口頭説明や論文における表現について改善例を伴ってご指摘頂いたこと，作図に関して貴重なご指導を頂いたことは本研究を進める上でも，本論文をまとめる上でもたいへん有益でした．心より感謝致します．

奈良先端科学技術大学院大学 情報科学研究科 中小路 久美代 助教授には，本研究を進めるにあたり，貴重なご助言，多くの関連研究をご教示頂きました．頂いたご助言は，本研究の発端となるものであり，たいへん有益でした．また，博士後期課程への進学に関して貴重なご助言を頂いたこと，英語表現においてご指導，ご支援頂いたこと，及び，生活面においても多大なご配慮，ご支援頂いたことに深く感謝致します．

奈良先端科学技術大学院大学 情報科学研究科 島 和之 助手には，本研究を進めるにあたり，多くのご助言，貴重なご指摘を頂きました．特に，研究の初期段階において，研究内容の矛盾に関する鋭いご指摘を頂いたこと，及び，計算機の設定をはじめとして技術的な知識を多く頂きましたことに心より感謝致します．

奈良先端科学技術大学院大学 情報科学研究科 門田 暁人 助手には，本研究の全過程を通じて，休，平日，昼夜を問わず，貴重なご助言，激励を頂きました．博士前期課程においては先輩として，論文の書き方，スライドの書き方や学生生活，計算機の設定に至るまで懇切丁寧なるご助言を頂きました．博士後期課程においては，本研究の方針，目的から論文の枝葉末節に至るまで有用なご指導，ご助言を頂きました．私が行きづまったときに，気分転換のための取計らいをご用意下さり，常に心強いご助言を頂きました．深く感謝致します．

本研究を進めるにあたり，奈良先端科学技術大学院大学 情報科学研究科 高田 眞吾 助手（現 慶應義塾大学 理工学研究科 専任講師），高田 義広 助手（現 オムロン株式会社），蔵川 圭 助手に心より感謝致します．高田 眞吾 助手には，博士前期課程，博士後期課程 1 年次において英語表現，プログラミングに関する貴重なご指導を頂きました．高田 義広 助手には，博士前期課程において，論文執筆上の注意，計算機設定，プログラミングに関して細部にわたり，ご支援，ご助言を頂きました．蔵川 圭 助手には，博士後期課程 3 年次において，博士課程学生としての姿勢を説示頂き，激励頂きました．

奈良先端科学技術大学院大学 情報科学研究科 白石 裕美 氏には，同じ研究グループのメンバとして本研究の方向，システム開発，評価実験に関して多くのご協力，熱心な議論をして頂きました．支援システムの機能実行履歴提示部分には，白石 氏のアイデアが多く含まれており，ユーザに対する効果的な機能実行履歴の提示が可能となりました．ま

た、無理なお願いでも快く聞いて頂きました。特に、システム開発においては、突発的な要求にも夜を徹して対応して頂きました。深く感謝致します。

奈良先端科学技術大学院大学 情報科学研究科 大和 正武 氏には本研究の全過程を通じて熱心に討論頂き、特に実装に関する多くのご助言、関連研究をご教示頂きました。本研究のアイデア段階の話を何度も聞いてもらい、いくつもの有用な指針を示して頂きました。また、システム開発、機能実行履歴の収集において多大なる貢献を頂きました。研究以外の面においても、苦手な酒につきあって頂いたこと、貴重な時間を潰してまでつまらない話に耳を傾けて頂いたことに深く感謝致します。

奈良先端科学技術大学院大学 情報科学研究科 山本 恭裕 氏には、本研究の関連研究をご教示頂くとともに、わかりやすい説明を加えて頂きました。博士後期課程進学に際して、悩みや喜びを共有することができ、非常に心強かったです。また、同期として国際会議などの多くの経験を共にしたこと、深夜、早朝に至るまで研究のあるべき姿や夢を共有したことにより、本研究を進める上での活力を得ることができました。深く感謝致します。

奈良先端科学技術大学院大学 情報科学研究科 鈴木 孝弘 氏（現 富士通株式会社）、菅原 千穂子 氏（現 シャープシステムプロダクト 株式会社）には、同期として様々なご援助、ご協力を頂きました。鈴木 孝弘 氏には本研究のデモンストレーションを見て頂き、新たな視点からご助言を頂きました。菅原 千穂子 氏には、本研究の他分野への適用について議論して頂きました。

奈良先端科学技術大学院大学 情報科学研究科 大平 雅雄 氏、松矢 明宏 氏、古賀 健太郎 氏、内田 眞司 氏、森川 昌平 氏（現 株式会社 SRA）に感謝致します。大平 雅雄 氏には、国際会議の準備をはじめとして、研究活動に伴う避けて通れない多くの作業を自ら快く引き受けて下さいました。また、精神面でも多くのご支援を頂きました。松矢 明宏 氏には、本研究を進める際に多くの激励を頂きました。無理なお願いも自ら快く引き受けて頂き、徹夜覚悟で待機して頂いたこともありました。また、精神面でも多くのご支援を頂きました。森川 昌平 氏、古賀 健太郎 氏には、本研究の礎を築く際にソフトウェア開発において多大なご協力を頂きました。内田 眞司 氏には無理なお願いばかりでしたが、全て快く引き受けて下さいました。

評価実験に御協力頂いた被験者の皆様、ソフトウェア計画構成学講座の皆様に感謝致します。

最後に、多くの面で支えてくださった両親に感謝致します。

付録

機能実行履歴の部分系列の抽出

全てのユーザの機能実行履歴の集合 \mathcal{O} から出現頻度 1 以上の機能実行履歴の部分系列の集合 \mathcal{P} を求め、機能実行履歴の部分系列のユーザ毎の出現頻度 \mathcal{F} を調べる。但し、出現頻度を調べる部分系列の最大長を設定する。

$o_{i,h}$ をユーザ i の機能実行履歴の h 番目の機能、 $P_{j,k}$ を長さ j の 1 つの機能実行履歴の部分系列、 $f_{j,k,i}$ をユーザ i の機能実行履歴における $P_{j,k}$ の出現頻度とし、 \mathcal{O} から \mathcal{P} 、 \mathcal{F} を、次のように求める。

全てのユーザの機能実行履歴の集合 \mathcal{O} :

$$\mathcal{O} = \{O_1, \dots, O_i, \dots, O_t\} \quad (t: \text{ユーザ数}) .$$

ユーザ i の機能実行履歴 O_i :

$$O_i = o_{i,1}, \dots, o_{i,n_i}$$

(n_i : ユーザ i の機能実行履歴の長さ) .

長さ j の機能実行履歴の部分系列の集合 P_j :

$$P_j = \{P_{j,1}, \dots, P_{j,k}, \dots, P_{j,l_j}\}$$

(l_j : $P_j \in \mathcal{O}$ である $P_{j,k}$ の数) .

長さ j の機能実行履歴の部分系列の集合 P_j のある要素 $P_{j,k}$:

$$P_{j,k} = \langle p_{j,k,1}, \dots, p_{j,k,j} \rangle$$

(ただし、 $1 \leq s \leq n_i - j + 1$ を満たすある s に対して $p_{j,k,1} = o_{i,s}$, $p_{j,k,2} = o_{i,s+1}$, \dots , $p_{j,k,j} = o_{i,s+j-1}$) .

あるユーザ i の機能実行履歴 O_i に含まれる $P_{j,k}$ のおける出現頻度を $f_{j,k,i}$ ($1 \leq i \leq t$) とし、

$$F_{j,k} = \langle f_{j,k,1}, \dots, f_{j,k,i}, \dots, f_{j,k,t} \rangle$$

長さ j の機能実行履歴の部分系列の出現頻度の集合 F_j :

$$F_j = \{F_{j,1}, \dots, F_{j,k}, \dots, F_{j,l_j}\} .$$

機能実行履歴の部分系列の最大長を m として、機能実行履歴の部分系列の集合 \mathcal{P} 、機能実行履歴の部分系列の出現頻度の集合 \mathcal{F} :

$$\mathcal{P} = \{P_1, \dots, P_j, \dots, P_m\}$$

$$\mathcal{F} = \{F_1, \dots, F_j, \dots, F_m\}$$

表 11 は抽出する部分系列の数 m を 2 とし、2 人のユーザの機能実行履歴をそれぞれ $O_1 = \langle 1, 2, 4, 3 \rangle$ 、 $O_2 = \langle 1, 4, 3, 1 \rangle$ とした場合の例である。

表 11 機能実行履歴と抽出する機能実行履歴の部分系列の例

P_1	$P_{1,1} = \langle 1 \rangle, P_{1,2} = \langle 2 \rangle, P_{1,3} = \langle 3 \rangle, P_{1,4} = \langle 4 \rangle$
P_2	$P_{2,1} = \langle 1, 2 \rangle, P_{2,2} = \langle 1, 4 \rangle, P_{2,3} = \langle 2, 4 \rangle, P_{2,4} = \langle 3, 1 \rangle, P_{2,5} = \langle 4, 3 \rangle$
F_1	$F_{1,1} = \langle 1, 2 \rangle, F_{1,2} = \langle 1, 0 \rangle, F_{1,3} = \langle 1, 1 \rangle, F_{1,4} = \langle 1, 1 \rangle$
F_2	$F_{2,1} = \langle 1, 0 \rangle, F_{2,2} = \langle 0, 1 \rangle, F_{2,3} = \langle 1, 0 \rangle, F_{2,4} = \langle 0, 1 \rangle, F_{2,5} = \langle 1, 1 \rangle$

f_u が 0 , もしくは , $f_s (1 \leq s \leq t, s \neq u)$ よりも十分小さい $P_{j,k}$ をユーザ u に提示する .
 $f_{j,k,u} (f_u \in \mathcal{F})$ の最低値を設けることにより (頻度が低い実行系列を枝刈りすることにより) \mathcal{P} の総数を小さくすることも可能である .

部分系列の実行頻度一覧には以下が含まれる .

- 頻度 ($\sum_{i=1}^t f_{j,k,i}$) : 全てのユーザの機能実行履歴における出現頻度
- 部分系列の長さ (j) : 部分系列を構成する機能数
- ユーザ毎の出現頻度 ($f_{j,k,1} \cdots f_{j,k,t}$)