

User Interface Dependability through Goal-Error Prevention

Robert W. Reeder and Roy A. Maxion. Proceedings of the 2005 International Conference on Dependable Systems and Networks, pages 60-69, Yokohama, Japan, 2005

Introduction

ユーザーインタフェースとヒューマンエラー

- ユーザーインタフェースは、ユーザーとコンピュータシステムの間、情報をやりとりすることに重要な役割を果たしている。
- しかし同時に、ユーザーインタフェースも、コンピュータシステムにおける脆弱性の一つの原因である。
- 信頼性の低いインタフェースは、ユーザーがタスクを完了する速さと正確さに関する仕様を満たすことができない。
- その原因の一つは、ヒューマンエラーである。
- ユーザーがエラーをする状況を解消するようにインタフェースを設計する場合、その信頼性はより高くなることができる。
- 従って、デザイナーはヒューマンエラーのタイプとその原因を理解し、ヒューマンエラーを防止する方法を見つける必要がある。

コンピュータセキュリティー

- ユーザーインタフェースの信頼性が極めて重要である一つの領域は、コンピュータセキュリティーである。
- 正しくないセキュリティー設定は、情報漏洩や外部攻撃などに至る危険がある。
- 今まで、私たちの研究は、Microsoft Windows XP におけるファイルパーミッションを設定する際のヒューマンエラーとユーザーインタフェースの信頼性を中心にしている。
- たくさんの事例証拠により、ファイルパーミッションを設定するのは、専門家ではない一般ユーザーにとって、エラーをしやすいタスクだと考えられる。

本研究の目的

1. 一般ユーザーがMicrosoft Windows XP でファイルパーミッションを設定する際のエラーを検討し、エラーが発生する原因を解明する。
2. ヒューマンエラーを減らすために、Salmonと呼ばれる新たなインタフェースを開発する。
3. 実験室で実験を行い、Microsoft Windows XPのファイルパーミッション設定のインタフェース（以下、XPFPと呼ぶ）と開発したSalmonというインタフェースを比較し、ヒューマンエラーを減らすに関してSalmonの有効性を確認する。

背景知識

- Microsoft Windows XP のファイルシステムは、アクセス制御リストモデル（Access Control List, 略してACL）を用いて、あるユーザーまたはユーザーグループがファイルやフォルダに対するアクセス権を設定する。
- ACLは、高度なアクセス許可を13通り定義した（ファイルの作成/データの書き込み、ファイルの一覧/データの読み取り、削除など）。
- ACLは、13通りのアクセス権を分けて六つの組み合わせになり、基本のアクセス許可と呼ぶ。
- それは、フルコントロール、変更、読み取りと実行、読み取り、書き込みと特殊なアクセス許可

- 例えば、「読み取り」という組み合わせは、四つの読み取りに関する高度なアクセス許可を含んでいる。
- これらのアクセス許可に対して、「許可」、「拒否」及び「セットなし」三つの状態がある。
- 「許可」より、「拒否」の方が優先される。
- 例えば、グループ1はあるファイルに対して、「書き込み」の許可権を持っている。
ユーザー1は、グループ1のメンバーとして、「書き込み」の許可権を与えられる（継承）。
この場合、ユーザー1に対して「拒否」を設定すると、「拒否」の方が優先され、ユーザー1はこのファイルに書き込むことができなくなる。
- 「セットなし」の場合は、すなわち「許可」も「拒否」も設定されていない場合、アクセス権は「拒否」として扱われる。
- 継承と優先ルールのため、ACLリストで表示されたアクセス権と実際に有効なアクセス権が異なる場合が存在している。

XPFPのインタフェース

- メインウィンドウ (Figure 1) で、上側はユーザーまたグループのリスト、下側は基本のアクセス許可についての設定である。
- 「Advanced」(詳細設定)をクリックすると、高度なアクセス許可を詳しく設定できる。
- また、「Advanced」のウィンドウに、「有効なアクセス許可」というタブがあり、そこであるユーザーまたグループが実際に持つ権限を見ることができる。

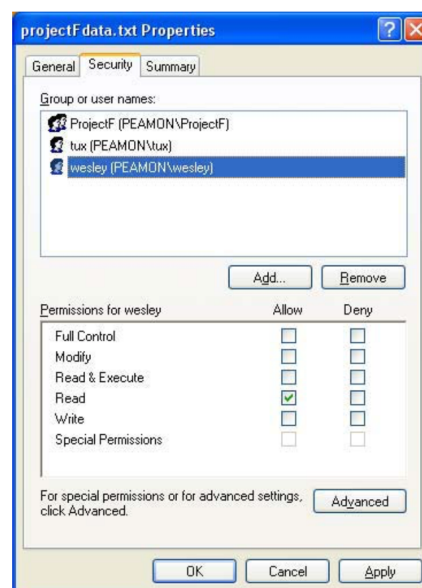


Figure 1: The XPFP interface. The interface contains information and functionality for setting permission values, but effective permissions, group-inherited permissions, and ADMINISTRATE permissions are not visible.

問題点

- メインウィンドウに、有効なアクセスが表示されていない。
- それをチェックしたい場合、ユーザーは「Advanced」、「有効なアクセス許可」をクリックし、新しい画面に移動する必要がある。

- 1回目は一人のユーザーまたは一つのグループの有効なアクセスだけを見ることができる。
- アクセス許可を変更する画面に戻ると、「有効なアクセス許可」の画面が消える。
- 従って、XPFPのユーザーは、設定されたアクセス権と有効なアクセス権の区別を理解する必要がある。
- また、「有効なアクセス許可」の画面がリアルタイムで提示されていないため、ユーザーは頭の中でアクセス権の継承と優先ルールを維持する必要がある。
- また、XPFPのインタフェースで、あるユーザーが所属しているグループの情報は提示されていないくて、その検索もできない。
- ユーザーはどのグループからアクセス権を継承したかについての情報もない。
- 従って、アクセス権を設定するとき、対象のユーザーが所属しているグループの情報を覚える必要がある。
- さらに、「特殊なアクセス許可 (Special permissions in figure1)」の中で、「アクセス許可の変更」という権限がある。
- この権限を許可されるユーザーは、自分のアクセス権を変えられる。
- 例えば、「書き込み」権を拒否されても、「アクセス許可の変更」を許可されるユーザーは、自分の「書き込み」権を許可することを通して、書き込むことを可能にする。
- 従って、ユーザーは、「詳細設定」の画面で隠されている「アクセス許可の変更」を理解する必要がある。

インタフェース改善

- ユーザーは、ファイルパーミッションに対する正しくないもしくは不十分なメンタルモデルを持っているかもしれない。
- これらのユーザーは、ファイル権限を設定する際に、インタフェースから新しい情報を取得し、メンタルモデルを更新する。
- 従って、インタフェースは、ユーザーがメンタルモデルを正しく更新するための情報を、受け取りやすいように提供する必要がある。
- ファイルパーミッションの設定にとって、必要な情報は以下。
 - ・ 13通りのアクセス許可のリスト
 - ・ グループメンバーの情報
 - ・ ユーザーのアクセス権と所属しているグループのアクセス権がどのように組み合わせるか
 - ・ 有効なアクセス権
- XPFPは、以上述べた情報を隠す、もしくは提供しない。
- 本研究は、新たなインタフェースSalmonを提案し、この問題を改善することを目指している。

Salmonインタフェース (Figure 2)

- Salmonは、「特殊なアクセス許可」のような組み合わせの代わりに、13通りの高度なアクセス許可を全部示す。
- インタフェースの上側で、アクセス許可を設定する。
- 下側で、有効なアクセス許可を表示する。
- 上側でアクセス許可の設定により、有効なアクセス許可はリアルタイムで下側で提示される。
- 下側で、あるユーザーを選択すると、このユーザーが所属しているグループの情報、またグループが持つアクセス許可の情報も同時に提示されている。
- 下側で、スクロールバーを使うと、同時に複数のユーザーの情報を見ることができる。

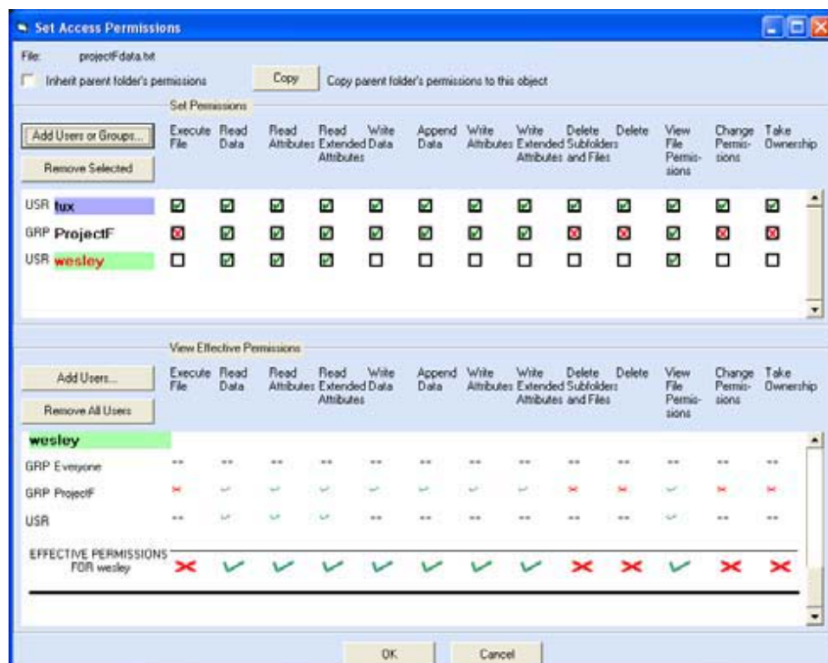


Figure 2: The Salmon interface. The upper pane of the Salmon interface contains the same information and functionality that is contained in the XPFP main window. The lower pane contains an effective-permissions display.

実験概要

実験参加者：

- カーネギーメロン大学の大学生と研究員 24 名
- 参加者全員は、windowsを使う経験がある。
- 19名の参加者はファイルパーミッションを設定したことがあるが、この領域の専門家でない。

実験機材と刺激：

- 参加者全員はWindows XP 2002を搭載した一つのパソコンで実験を行う。
- Windows XP 本来のXPFPインタフェースと提案したSalmon、二種類のインタフェースを利用してタスクをする。
- 実験中、思考発話法が利用され、参加者の発話を集める。
- ユーザーの操作は、あるユーザー調査のためのソフトウェアで記録される。

タスク：

シナリオ：

- 実験参加者は、ある会社で働いていて、他の社員と一つのパソコンを共用する。
- このパソコンで、実験参加者を含んで27名のユーザーがあり、ProjectA~Fという六つのグループに所属している。
- 一つのグループは、6名のユーザーを含んでいる。
- 実験参加者は、管理者として、他のユーザーのファイルパーミッションを管理している。

- 実験参加者は、それに関する七つのタスクをするように要求される。
- その中の二つのタスク、WesleyタスクとJackタスクはデータ分析の対象になる。

Wesleyタスク：

- ・ 会社で、グループProjectFは、「ProjectFdata.txt」というファイルを取り扱っている。
 - ・ グループProjectFのすべてのメンバーはこのファイルを読み取る・書き込む・削除することができる。
 - ・ Wesley（ユーザーの名前：Wesley）は、人事異動のため、ProjectFから別のグループに配属された。
 - ・ 今、WesleyはまだグループProjectFの一員である。
 - ・ 彼をファイルは読み取れるが、その内容を変えることはできないように設定してください。
- Jackタスクの説明でも、グループの名前とファイルの名前以外の他の内容はWesleyタスクと同じ
 - JackはProjectEから別のグループに異動し、ProjectEの一員として「ProjectEdata.txt」というファイルの内容を変えることができなくなる。
- WesleyタスクとJackタスクの区別は、グループProjectFとProjectEのアクセス許可が異なる。
 - ProjectFで、「書き込み」と「読み取り」は許可されていたが、「アクセス許可の変更」は拒否されていた。
 - ProjectEで、「書き込み」、「読み取り」と「アクセス許可の変更」を許可されていた。
- Wesleyタスクでは、Wesleyに「書き込む」の拒否権を与えるとタスクに成功できる。
 - しかし、Jackタスクでは、「書き込み」以外に、Jackの「アクセス許可の変更」を拒否する必要がある。

タスクルール：

- 実際の使用場面をシミュレーションすると同時に、二つのインタフェースを比較するために、タスクに関するルールを設定した。
- XPFPインタフェースで、グループのメンバーについての情報を検索できない。
 - そのファンクションは、システムの「コンピュータの管理」に関するインタフェースにある。
 - 実験参加者が「コンピュータの管理」インタフェースを使って、グループとユーザーの情報を検索する行為は許される。
 - しかし、グループのメンバーを変えることは禁止される。
 - 例えば、グループProjectFからWesleyを削除する操作が認められない。
 - このルールにより、参加者がシステムの状態を変更する操作をファイルパーミッションのインタフェースに制限する。
- グループのメンバーの変更に関する制約があるため、タスクの説明で言及されていないユーザーのアクセス権の変更は認められる。
 - 従って、他のメンバーのアクセス権にも影響するが、参加者はグループProjectE,Fのアクセス権を変えることで、WesleyとJackのアクセス権を設定することができる。
- 参加者は、Windows XPのファイルパーミッションについてのオンラインヘルプファイルを参考できる。

手順：

- まず実験参加者に「コンピュータの管理」インタフェースを使ってユーザーとグループを検索する方法を示す。
- 次に、Windows XPのファイルパーミッションについてのヘルプファイルにオンラインでアクセスする方法を示す。
- XPFPとSalmonについての説明はない。
- タスクは7つあり、その説明はテキストで、ウェブブラウザで提示される。
- 参加者全員は、まず一つのトレーニングタスクをする。
- 残りの六つのタスクは、参加者の間にカウンターバランスを取るよう提示される。
- 参加者は8分以内一つのタスクを完了するように要求される。（専門家は1分以内にタスクを完了できる）

結果分析：

- 分析対象：Wesleyタスク、Jackタスク
- 分析データ：タスクの結果（成功・失敗）、完了時間とエラーの数
- 分析は、以下五つの段階がある。

1. タスクの成功・失敗を確認する

- WesleyとJackが持つ有効な「書き込み」「読み取り」「アクセス許可の変更」に関する権限をチェックすることによって、タスクの結果を確認する。
- この三つ以外のアクセス権の設定状態は考えられない。

2. HTA図を作る

- エラーを確認するために、階層的課題分析法（hierarchical task analysis, HTA）を用いて、WesleyタスクとJackタスクの正しいやり方を図で表現する。
- HTAは、タスクを階層的に分解する方法である。
- JackタスクのHTA図はFigure 3で示される。
- HTAにより、Jackタスクは「読み取り」についてのサブゴール1、「書き込み」についてのサブゴール2、「管理（アクセス許可の変更）」についてのサブゴール3に分解する。
- その後、各サブゴールを実現できるすべての計画を立てる。
- 計画により、すべき操作を順番に書く。

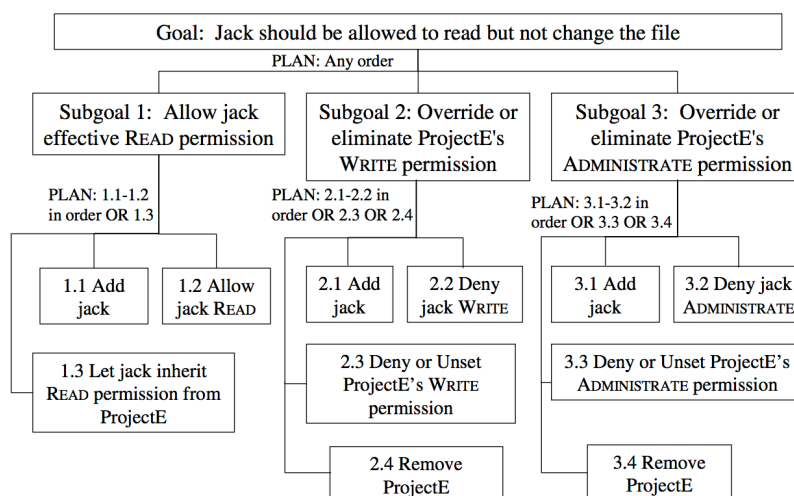


Figure 3: Hierarchical Task Analysis of the Jack task.

3. 実験参加者の操作リストを作る

- HTA図を比較するために、ユーザーの操作を確認する必要がある。
- 本研究で、操作を、アクセス許可リスト (ACL) の変更と定義する。
- すなわち、ACLにエントリを追加する操作、エントリを削除する操作と編集する操作。
- この定義により、タスク中の実験参加者の操作リスクを作る。

4. エラーを確認する

- 参加者の操作をHTAの結果と対比し、ヒューマンエラーであるかどうかを確認する。
- HTAにより、実験参加者の操作とHTA図の操作の違いを、オMISSIONエラー (omission error)、コミッションエラー (commission error) と非エラーに分類する。
- オMISSIONエラー・省略エラーは、すべきことをしない行為。(HTA図にある操作をしないこと)
- コミッションエラーは、やってはいけないことをした行為。(HTA図にない操作をしたり、HTA図にある操作を正しくない順番に、もしくは正しくないタイミングでしたりすること)
- 非エラーは、必要ではないが無害なことをした行為。
- 例えば、アクセス権を変更して、何が発生したかを確認した後、元の状態に戻す行為は非エラーである。

5. エラーを分類し、その数を計算する

- 先行研究により、Human Error Assessment (THEA) (Pocock et al., 2001) を用いて、エラーを分類する。
- THEAは、ノーマンのユーザー行為に関する7段階モデルより開発され、人間のインタフェース行為を目標実現行動として捕らえて、四つの段階 (知覚・解釈・評価段階、目標設定段階、計画段階、実行段階) にまとめる。
- 知覚・解釈・評価段階で、ユーザーはシステムから情報を取得し、解釈して、目標を実現したかどうかを判断する。
- 目標をまだ達成していない場合、ユーザーは目標設定段階に入り、目標を実現するためのサブゴールを作る。
- 計画段階で、ユーザーはサブゴールを実現するための計画を立てる。具体的に、操作系列を選択し、その順番を決める。
- 最後、計画での操作を実行する。
- エラーが発生した段階により、四つの種類、目標エラー、計画エラー、実行エラー、知覚・解釈エラーに分類する。
- 例えば、あるエラーを目標エラーに分類する場合は以下：
ユーザーが必要なサブゴールを設定しないため、オMISSIONエラーが発生した。
ユーザーが正しくないサブゴールを設定したため、コミッションエラーが発生した。
- サブゴールが設定されたかどうかは、主に実験参加者の発話より判断する。
- Wesleyタスクで、よくある目標エラーは、Wesleyの「書き込み」権を拒否しなかったこと。
- Jackタスクで、よくある目標エラーは、Jackの「書き込み」権を拒否しなかったこと (Figure3のsubgoal2を省略したこと) と、Jackの「アクセス許可の変更」権を拒否しなかったこと (Figure3のsubgoal3を省略したこと)。

結果：

タスク完了時間 (Figure 4)

- Wesleyタスクで、XFPFグループとSalmonグループの参加者全員の平均時間を比較して、結果はほぼ同じである。
- Jackタスクも同じような結果が見られた。

- しかし、XPFPグループの中で、すべき操作をしなかったため、タスクに失敗した参加者はたくさん存在している。
- このような参加者は、タスクの完了時間が短くなる傾向がある。
- 従って、二つのグループの中で、タスクに成功した参加者の完了時間を比較するのは、意味あると考えられる。
- 結果として、WesleyタスクとJackタスクで、Salmonグループは、XPFPグループよりタスク完了時間が短い、有意な差はなかった。

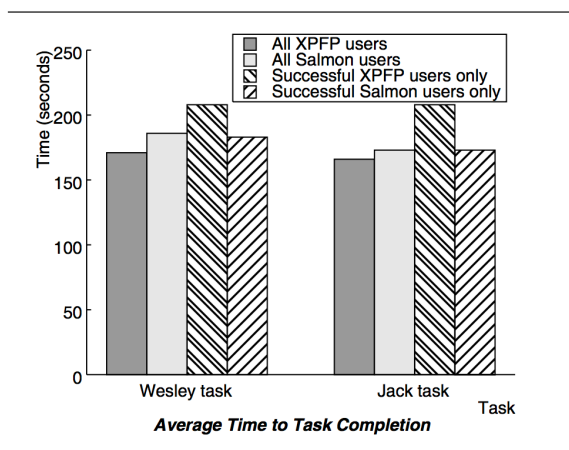


Figure 4: Average time to task completion for the Wesley and Jack tasks. Amongst successful users, Salmon moderately outperformed XPFP in time to task completion, but the differences were not statistically significant.

	Wesley task	Jack task
XPFP	58%	25%
Salmon	83%	100%

Table 1: Accuracy rates for the Wesley and Jack tasks on the XPFP and Salmon interfaces. Salmon showed 43% and 100% improvements in accuracy over XPFP on the Wesley and Jack tasks, respectively.

	Goal	Plan	Action	Perception
Wesley				
XPFP	5	1	0	3
Salmon	1	2	1	0
Jack				
XPFP	15	0	0	1
Salmon	1	3	2	0

Table 2: Count of errors by type for Wesley and Jack tasks on XPFP and Salmon interfaces. Salmon users committed significantly fewer goal errors than did XPFP users for both the Wesley and Jack tasks.

タスクの結果 (Table 1)

Wesleyタスク :

- XPFPグループの12名の参加者の中で、7人 (58%) はタスクに成功した。
- Salmonグループの12名の参加者の中で、10人 (83%) は成功した。
- z検定により、SalmonグループとXPFPグループの間に、有意な差が見られた ($z=1.347$, $p=0.089$)。

Jackタスク :

- XPFPグループの12名の参加者の中で、3人 (25%) はタスクに成功した。
- Salmonグループの12名の参加者の中で、12人 (100%) は成功した。
- z検定により、SalmonグループとXPFPグループの間に、有意な差が見られた ($z=3.795$, $p < 0.0001$)。
- XPFPより、Salmonは、タスクの正確さに優れている結論が出られる。

ヒューマンエラーの数 (Table 2)

Wesleyタスク :

- XPFPグループの参加者は、合計九つのエラーをして、その中の五つが目標エラーである。
- Salmonグループの参加者は、合計四つのエラーをして、一つが目標エラーである。
- z検定により、SalmonグループとXPFPグループの間に、目標エラーの数は有意な差が見られた ($z=-1.885$, $p=0.0297$)。

Jackタスク :

- XPFPグループの参加者は、16のエラーをして、その中の15個が目標エラーである。
- Salmonグループの参加者は、六つのエラーをして、一つが目標エラーである。
- z検定により、SalmonグループとXPFPグループの間に、目標エラーの数は有意な差が見られた ($z=-3.312$, $p=0.0005$)。

- エラー分析により、二つのタスクでも、Salmonインタフェースを使う参加者は、XPFPより少ない目標エラーをしたことが示された。
- また、Salmonグループの参加者は、XPFPより多くの計画と実行エラーをした。
- しかし、本研究で、ほとんどの目標エラーはタスクの失敗に直接関連しているが、計画エラーと実行エラーは、タスクの成功に大きな影響がない。
- さらに、ほとんどの計画エラーと実行エラーは、タスク中修正された。
- しかし、Salmonの計画エラーと実行エラーをさらに検討する必要がある。

ディスカッション

- 実験を通して、Salmonがタスクの正確さの改善と目標エラーの減少に対する有効性は証明された
- Salmonインタフェースに、ビジュアルデザインの変更（例えば、フォントの種類、色、タブ、レイアウトなど）はあるが、実験参加者の発話によって、正しいサブゴールを設定できるのは「有効なアクセス許可」についての情報が提示されるためである。
- 例えば、実験中、あるSalmonを使ってる参加者は、Jackタスクの正しくない解決策を提出したい時、下側の有効なアクセス許可の提示に気がついた後、自分がエラーをしたことがわかった。
- それに対して、XPFPを使ってる参加者は、Figure 1でWesleyの「読み取り」の許可 (Allow) がチェックされて、「書き込み」の許可 (Allow) がチェックされていないのを見た後、Wesleyが読み取れるが書き込めないことを信じていた。

総合考察と今後の課題

- 本研究は、Windows XPシステムでファイルパーミッションを設定する際に利用したユーザーインタフェースの信頼性とヒューマンエラーを検討した。
- 研究の対象ユーザーは、専門家ではない一般のユーザーである。
- 私たちは、一般ユーザーも、自分の情報を守るニーズと必要があると考えられるため、これらのユーザーの立場で信頼性が高いインタフェースを提案することを目指していた。
- 実験を通して、提案したSalmonインタフェースを使う一般ユーザーは、より少ない目標エラーをして、より正しくタスクをできることが証明された。
- 今後は、感知・解釈エラー、計画エラーと実行エラーを減らすようにインタフェースを改善する課題に着目する。