

Human Interface Guideline に基づく GUI のユーザエラー防止性自動評価手法

片桐 健吾[†] 小形 真平[‡] 岡野 浩三[†]

信州大学工学部情報工学科[†]

信州大学大学院理工学研究科[‡]

1. はじめに

アプリケーションが利用者の誤操作を軽減できるような構成されることは重要であり、この質はユーザエラー防止性として扱われる。本特性の向上手段の一つに、ボタンなどの GUI 要素を適切にレイアウトすることが挙げられる。そして、適切な GUI レイアウトは要求確認プロセスにて利用者に操作イメージを正確に伝えるために必要であり、開発の初期段階で良質な GUI レイアウトを効率的に得ることが求められる。

GNOME HIG (Human Interface Guideline) (以降、単に HIG) は GUI 要素のレイアウト方法を客観的かつ具体的にガイドする項目 (以降、ガイド項目) を多量に含むガイドラインである。しかし、開発者は多量のガイド項目を全て手動で扱う必要があり、GUI レイアウトがガイド項目を満たすかどうかの評価に時間がかかる。

本稿では、評価の効率化のため、HIG に基づいて GUI レイアウトを自動評価することによるユーザエラー防止性自動評価手法を提案する。提案手法では、先行研究の上流工程向けのレイアウトツール [1] をベースに得られるレイアウトを評価対象とする。そして、手法実現のために、GUI 要素の種類やサイズ、GUI 要素間の関係から評価できるガイド項目を抽出し、その項目ごとに評価式を構築し、GUI レイアウトを自動評価するツールを実現する。

提案手法を既存のアプリケーションから再現した GUI レイアウトに適用した結果、修正可能な違反を発見できたことから、提案手法が有効である見込みを得た。

2. 用語説明

2.1. ユーザエラー防止性

JIS X 25010 によるとユーザエラー防止性は“利用者が間違いを起こすことをシステムが防止する度合い”と定義される。

不適切な GUI レイアウトが認知負荷の観点からユーザの誤操作を誘発しうる [2] ことから、本

稿では、GUI レイアウトの観点に絞ってユーザエラー防止性を捉えるものとする。

2.2. GNOME Human Interface Guideline (HIG)

HIG は、開発者やデザイナーに関わらず GTK+ を用いた有用なアプリケーションの設計するための原則である。GTK+ は他のツールキットが含む GUI 要素を多数含むことから、HIG の汎用性は低くない。また、実装技術としての GTK+ は、GUI 要素の整列方法を一部提供するが、上流工程への利用には向かない。

HIG は GUI 要素のサイズや位置関係、テキストの記述方法、イメージやアイコンの使い方、コンテナ (グループ) 内の GUI 要素の位置関係や設置数などのガイド項目を含む。例えば、ラジオボタンのグループには「同じ見出し下のグループの中で 8 つ以上ラジオボタンを設置しない」というガイド項目がある。本項目を満たすことで、認知負荷によるユーザの誤操作を軽減できるであろう。すなわち、HIG を満たすことでユーザエラー防止性が向上すると期待される。

3. 提案手法

本研究で実現する自動評価ツールを中心に提案手法を説明する。提案ツールへの入力は、3.1 節の GUI レイアウトとなる。出力は、ガイド項目に基づく評価項目を GUI レイアウトが満たしたかどうかを項目ごとに評価した結果である。

3.1. GUI レイアウト

GUI レイアウトは、Page の名前、Widget の位置やサイズ、Widget をまとめた Group の位置やサイズを含み、Widget や Group には表 1 に示す種類がある。本レイアウトは、先行研究のレイアウトツール [1] をベースに得られる。

3.2. HIG に基づく自動評価

HIG には多量のガイド項目がある。本研究では、HIG の原文を箇条書きや文から、さらに細かい箇条書きに分割したものを評価項目と呼ぶ。

そのように得た項目数は 274 個であり、その内 22 個を計算可能な項目として抽出した。計算可能な基準はサイズや位置、個数などの数値表現や、「隣の項目」や「同じ幅」などの数値的に捉えられる表現でガイドされるものである。

表 2 に計算可能とした評価項目の例を示し、表 3 に表 2 に対応した、○ (満たす) か × (満

Automated Evaluation of GUI based on
Human Interface Guideline for User Error Protection

[†]Kengo Katagiri, [‡]Shinpei Ogata, [†]Kozo Okano

[†]Department of Computer Science & Engineering, Faculty of Engineering, Shinshu University.

[‡]Graduate school of Science and Technology, Shinshu University.

表 2 評価項目(一部)

No.	評価対象	評価項目
1	ボタン	隣にボタンがある場合、それらが同じ幅を持つようにする
2	ラジオボタン	ラジオボタンのグループは垂直方向に配置する
3	ラジオボタン	同じ見出し下のグループの中で8つ以上ラジオボタンを設置しない

表 3 表 2 に基づく条件式

No.	評価式
1	<code>baseButton.getWidth() == nextButton.getWidth()</code> baseButton:=サイズの比較元となるボタン nextButton:=比較される baseButton と x 座標の一致するボタン
2	<code>baseX == radioButtonGroup.getRadioButton().getX()</code> baseX:=x 座標の比較元となるラジオボタン radioButtonGroup.getRadioButton():=比較されるグループ内の1つのラジオボタン
3	<code>radioButtonNumber > 0 && radioButtonNumber < 8</code> radioButtonNumber:=グループ内のラジオボタンの数

表 1 Widget と Group の種類 (一部)

Widget	Group
BUTTON_PUSH	GROUP_CHECK
BUTTON_RADIO	GROUP_RADIO
BUTTON_CHECK	COMPOSITE
LABEL	
TEXT_SINGLE	

たさない) を計算して判定する条件式を示す。

4. 評価実験

実際のアプリケーションに対して提案手法が有効であるかどうか、つまり、違反が見つかるかどうかを評価した。適用対象は、複雑性を考慮し、図 1 に示す既存のリモートログオンクライアントの接続機能のプロキシ設定画面とした。

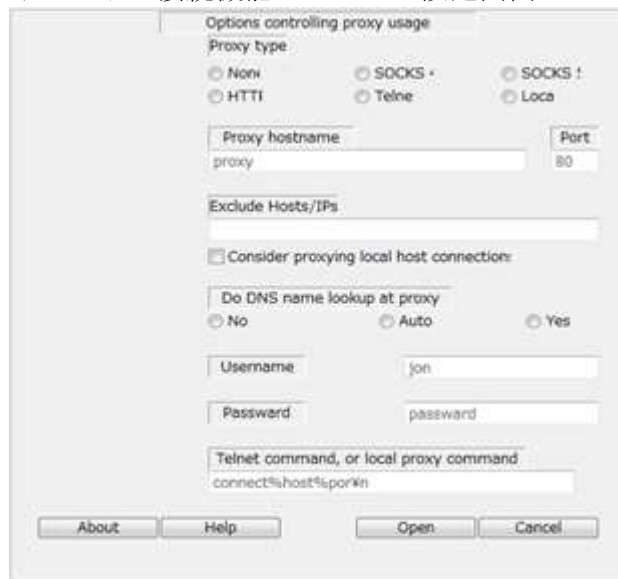


図 1 評価対象レイアウト

4.1. 評価結果

適用結果として、表 2 に示す範疇では No. 1 と 3 が「○」、No. 2 が「×」と出力された。

4.2. 考察

実在するアプリケーションで項目違反が存在し、それを検出できている。加えて、本違反はラジオボタンを垂直方向に配置し直すことで修正可能であり、実際に手動で修正できた。以上から、提案手法が有効である見込みが得られた。

5. まとめ

本稿では、HIGに基づいて GUI レイアウトを自動評価することによるユーザエラー防止性自動評価手法を提案した。実在するアプリケーションから修正可能な評価項目違反を検出したことから、提案手法が有効である見込みが得られた。

今後の課題として、計算可能かつ未実装な評価機能を実装する。また、多様なアプリケーションに提案手法を適用し、汎用性の評価と有効性のさらなる評価を行う。そして、HIGの違反がユーザエラー防止性に影響するかを裏付けるために、ユーザ操作による評価実験を計画する。

6. 謝辞

本研究は JSPS 科研費 15K15972 の助成を受けたものです。

参考文献

- [1] 赤瀬智也, 予測入力を用いた GUI レイアウト作成支援ツール, 平成 25 年度信州大学工学部情報工科学士学位論文, 2014.
- [2] E. Folmer, et al., A framework for capturing the relationship between usability and software, in Software Process: Improvement and Practice, vol. 8, no. 2, pp. 67-87, 2003.