

学生実験のための状態遷移図を用いたインタフェース作成支援ツール

A User Interface Design Tool Using the State Transition Diagram to Support Classes

金 秀男†
Xiunan Jin倉本 到†
Itaru Kuramoto渋谷 雄†
Yu Shibuya辻野 嘉宏†
Yoshihiro Tsujino

1. はじめに

近年、ソフトウェア開発においてユーザインタフェースの重要性が特に認識されるようになってきた。本学電子情報工学科においても、数年前から学部2回生を対象とした実験のテーマの一つとして、「ヒューマンインタフェース(HI)基礎実験」[1]が設けられており、講義と連動してHIに関する教育を行っているが、十分な実験時間を確保できないという問題がある。本稿では、この実験を効率的に遂行するために必要とされる支援機能について検討し、それに基づいて試作したツールを紹介する。

2. HI基礎実験の現状

HI基礎実験では、学生にPC上で、あるインタフェースを実現させ、そのユーザビリティを評価させている。学生が設計したインタフェースを実装する際には、一般のUI開発ツール[2]を用いても、プログラミングが必要である。本学の実験では、Microsoft Visual J++を用いている。

このような開発ツールは、簡単なインタフェースを作成する場合でも、コードを書かなければならない。そのため、学生にはプログラミングの知識が必要となる。また、学生がプログラミングの知識や技術をある程度持っていても、インタフェースの作成には多くの時間がかかる。さらに、作成したインタフェースを評価するためにはユーザの操作履歴を記録する機能を追加する必要がある。

以上のように、プログラミングを必要とする現状では、評価すべきインタフェースの作成に非常に時間がかかってしまう。そのため、インタフェースの設計、評価、および再設計のための時間が限られてしまう。

HI基礎実験の目的は、学生が自らインタフェースを作成し、実際に操作し、評価することにより、ユーザビリティを重視したインタフェース設計を学ぶことである。そこで、本支援ツールでは、実験の本来の目的である、インタフェースの設計と評価を通したユーザビリティの重要性の学習に集中できるように、プログラミングの複雑さをできるだけ排除することで、インタフェース作成を支援する。

3. インタフェース作成支援ツール

3.1 ツールが持つべき特徴

- インタフェース画面を短時間で簡単に作成できること
- ユーザの操作に対応するシステムのインタフェース上の動作と簡単な内部動作を通常のプログラミングを行うことなしに記述できること
- 以上の記述だけで実際にインタフェースが動くこと
- 動作したときのログが自動的に取れて、それを元にユーザビリティの分析が可能であること

3.2 試作ツール

以上の特徴を備えるツールを試作した。試作した支援ツールの概観を図1に示す。また、図2にインタフェース作成支援ツールを用いたインタフェースの作成手順を示す。

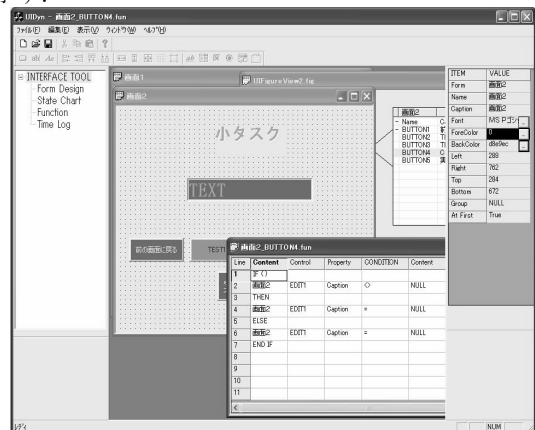


図1 ツールの全体

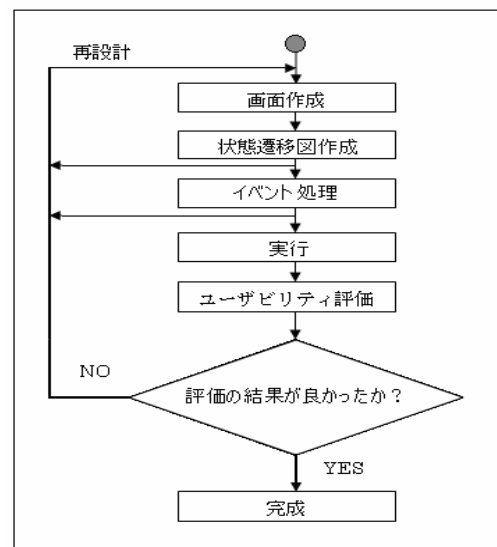


図2 インタフェースの作成手順

本ツールの主な機能である画面作成、状態遷移図作成、イベント処理設定、インタフェースの操作履歴記録について以下に説明する。

① 画面作成 (図3参照)

画面作成の際に必要な、外観を描画する機能と、画面にボタンなどのコントロールを配置する機能を備える。学生が直接画面上にコントロールを配置し、そのキャプション、位置、色などの属性を設定し、画面を作成する。なお、このような画面作成機能は従来のプログラム開発環境も備えている。

† 京都工芸繊維大学

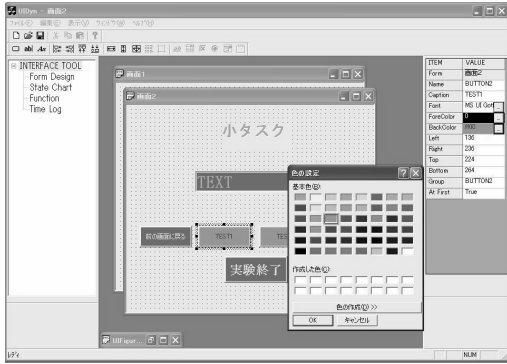


図3 画面作成

② 状態遷移図作成 (図4参照)

画面の切り替えは、状態遷移図によって記述し、プログラミングを必要としない。画面切り替え以外のインタフェース上の動作と簡単な内部動作は③に述べるイベント処理設定により作成する。

作成したインタフェース画面を保存後、状態遷移図を作成する。図4は、状態遷移図作成画面である。作成したインタフェース画面毎にその画面上にあるボタンのリストが表示されている。そのリストアイテムはボタンの名とキャプションが表示されている。あるリストのボタンアイテムを選択し、反転した領域を他のリスト画面にドラッグするとそのボタンとリスト画面の間に線が引かれ、状態遷移を指定できる。つまり、インタフェースの実行時には、そのボタンを押すと、ボタンと線で結びついた画面に切り替わることを意味する。

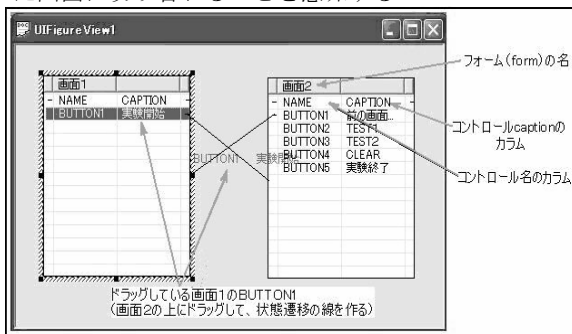


図4 状態遷移図作成

③ イベント処理設定

作成したインタフェース画面上のボタンを押すなどのイベントを処理する部分を容易に記述できる。イベント処理としては、インタフェース操作の終了、画面切り換え (状態遷移)、画面上のコントロールの属性変更、コントロールの属性に応じた条件分岐 (IF 文による記述) を行うことができる。

状態遷移図の画面上にあるリストのボタンアイテムをダブルクリックするとそのボタンのイベント処理の設定画面 (図5参照) が表示される。

インタフェース操作の終了は、“Content”欄に“EXIT”を設定することで指定する。

画面切り換えは、“Content”欄にインタフェース画面名、“Control”欄に“SHOW”あるいは“HIDE”を設定することで指定する。

画面上のコントロールの属性変更については、まず、“Content”欄にインタフェース画面名、“Control”欄にそのインタフェース画面上のコントロールの名前、

“Property”欄に対象となるコントロールの属性名を指定する。そして、“CONDITION”欄に代入あるいは追加演算子 (表1参照) を指定し、右側に値を直接書くか特定のコントロールの属性を指定すると左側に指定したコントロールの属性値を変更することになる。

IF文については、“Content”欄で“THEN”および“ELSE”を選択し、その次の行に行いたい操作を指定する。“Content”欄で“END IF”を選択するとIF文を終了する。IF文の“CONDITION”欄では比較演算子 (表1参照) を選択し、さらに、その右側の欄では比較相手のインタフェース画面名、コントロール名、および属性名を指定する。

例えば、図5のイベント処理では画面2でボタン4 (BUTTON4) を押したとき、テキストボックス (EDIT1) のキャプションが存在するなら、EDIT1のキャプションを消す、その以外の場合、EDIT1のキャプションを“文字がない”に変える処理を記述している。

表1 CONDITION欄で選択可能な演算子

演算子	説明
=	左側のキャプションに右側で設定した値をテキストとして代入する
==	両側の値が同じ
<>	両側の値が異なる
<=	左側のキャプションに右側で設定した値をテキストとして追加する
Back	左側のキャプションの最後の文字を一文字消す

Line	Content	Control	Property	CONDITION	Content	Control	Property
1	IF						
2	画面2	EDIT1	Caption	◇	NULL		
3	THEN						
4	画面2	EDIT1	Caption	=	NULL		
5	ELSE						
6	画面2	EDIT1	Caption	=	文字がない		
7	END IF						

図5 イベント処理の設定

④ インタフェースの操作履歴記録

作成したインタフェースを実行すると、その作ったインタフェースの操作によるイベントの種類およびそのイベントの発生時刻が自動的に記録されるので、学生はそれを利用して、ユーザビリティを評価できる。

4. まとめ

本稿では、HI基礎実験において、学生のプログラム作成時間を削減することで、短時間で学生にユーザインタフェースの作成、評価、再設計のサイクルを体験させるための支援ツールを試作した。これにより、ヒューマンインタフェース教育の効果を上げることが期待される。

参考文献

[1] 電子情報工学基礎実験 電子情報工学実験及び設計 I・II 2004・2005年, 京都工芸繊維大学 工芸学部 電子情報工学科, p89-p102 (2004) .
 [2] BRAD MYERS, SCOTT E. HUDSON, and RANDY PAUSCH: Past, Present, and Future of User Interface Software Tools , ACM Transactions on Computer-Human Interaction, Vol. 7, No.1, pp.3-28 (2000).