

ユーザインターフェース設計におけるインタラクション 仕様分析手法の提案

石山 啓太郎 木下 哲男 白鳥 則郎
東北大学電気通信研究所/情報科学研究科
〒980-8577 仙台市青葉区片平2丁目1番1号
TEL:022-217-5454

e-main:{k-taro,kino,norio}@shiratori.riec.tohoku.ac.jp

あらまし ユーザインターフェース(UI)の設計においてユーザビリティの確保は重要課題となっており、効率的、効果的な UI 開発には設計プロセスでの様々な局面におけるユーザビリティ分析・評価が必要となる。本稿では、ユーザ/タスク中心の設計方法論の観点から UI 設計プロセスにおけるインタラクション設計を効果的に支援するために提案しているタスク指向インタラクション設計記法 TID(Task-oriented Interaction Description technique)に基づいたユーザビリティ(インタラクションの一貫性)の分析手法を提案する。

Analysis Method of Specification of Interaction Specification In User Interface Design.

Keitaro Ishiyama Tetsuo Kinoshita Norio Shiratori
Research Institute of Electrical Communication / Graduate School of Information
Science, Tohoku University
2-1-1, Katahira, Aoba-ku, Sendai 980-8577
TEL:022-217-5454

e-main:{k-taro,kino,norio}@shiratori.riec.tohoku.ac.jp

Abstract It is important to make sure of usability of a user interface to be designed, to do so, it is required the designers to analyze usability of the user interface in various aspects of a design process of efficient user interface. In this paper, from a viewpoint of user/task oriented method, we propose a method to analyze usability (consistency of interaction) of user interfaces based on the TID method which supports the interaction design of user interface.

1. はじめに

ユーザインターフェース（以下 UI）の設計においてユーザビリティの確保は重要課題となっており、効率的、効果的な UI 開発には設計プロセスの様々な局面でのユーザビリティ分析・評価が必要となる。そこで、本稿では TID により記述された TID 仕様に基づくユーザビリティ分析手法の一つとしてインタラクションの一貫性を分析する手法を提案する。

2. TID 仕様について

ユーザ/タスク中心の設計方法論[2]に基づいた実際の UI 設計作業、特にインタラクション設計を効果的に支援するための手法として、タスク指向インタラクション設計記法 TID (Task-oriented Interaction Description technique) [1]による UI 設計手法を提案している。TID においてインタラクションはタスクを基本単位として記述され、タスク間に存在する関係を 12 個のタスク関係子（図 2）で表現していく形でタスクの詳細が設計される。UI システム全体のインタラクション設計では、タスクをサブタスクとタスク関係子によって定義し、その繰り返しにより仕様を階層的に記述することが可能となる。また、それ以上詳細化出来ないタスクに対しては具体的なユーザの操作を表わすユーザアクション(UA)を割り当てることになり、以下の形式で記述する。

UA: デバイス-動作(デバイスラベル)"対象オブジェクト"

例) UA: mouse-click(左ボタン)"アイコン"

TID 仕様ではユーザに提供される単位機能としての定義された"UI 機能"を軸として、個々の UI 機能へアクセスするための手順を記述した UI 機能アクセス手順記述部、個々の UI 機能を実行するための手

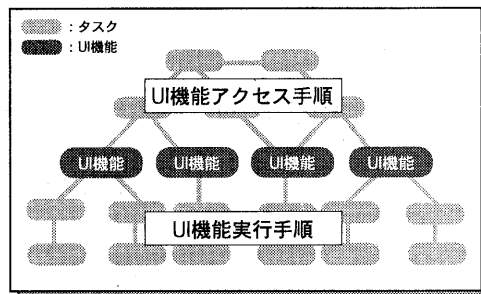


図 1:TID 仕様イメージ

順を記述した UI 機能実行手順記述部とに分けられる（図 1）。記述されるタスクは、それに対応する UI 機能に関する情報（どの UI 機能へ対するアクセス、もしくは実行手順の一部なのか）を持ち、これを UI 機能情報と呼ぶ。

3. 一貫性分析手法

UI のユーザビリティに影響する要因には、使いやすさ (ease of use)、学びやすさ (easy to learn)、透明度 (transparency) など様々なものが考えられる。そして、UI に一貫性を持たせるとそのシステムが使いやすくなることはほとんど自明の理であり、前記の学びやすさや透明度に影響を及ぼす設計要因の一つは一貫性と考えられる。したがって、UI における一貫性はユーザビリティを評価する際の有効な特性の一つであると考えられる。

タスク関係子	
記号	意味
>>	sequence
[>>	skippable
[>	interrupt
[><	modal interleave
><	order independence
	choice
F	parallel first
L	parallel last
*	0 回以上の再帰実行
+	1 回以上の再帰実行
n	n回 (定数回) の再帰実行
*	0 または 1 回の再帰実行

図 2:タスク関係子

```

開く:=ファイル選択 [> 決定
      UA:mouse-click"ボタン"
ファイル選択:=FI (アイコン選択, ファイル名入力選択)
アイコン選択:=ディレクトリ移動* [> アイコン選択
      UA:mouse-click"ボタン" UA:mouse-click"アイコン"
ファイル名入力選択:=入力選択 >> 入力*
      UA:mouse-click"入力エリア" UA:key-press" * "
      [> 決定
      UA:mouse-click"→入力エリア"
方形描画:=起点決定 >> 各パラメータ設定 [> 描画決定
      UA:mouse-click"フィールド" UA:mouse-click"ボタン"
各パラメータ設定:=ILI (幅データ入力, 高さデータ入力)
幅データ入力:=入力選択 >> 入力*
      UA:mouse-click"入力エリア" UA:key-press" * "
      [> 設定
      UA:mouse-click"→入力エリア"
高さデータ入力:=入力選択 >> 入力*
      UA:mouse-click"入力エリア" UA:key-press" * "
      [> 設定
      UA:mouse-click"→入力エリア"
円形描画:=起点決定 >> 各パラメータ設定 [> 描画決定
      UA:mouse-click"フィールド" UA:mouse-click"ボタン"
各パラメータ設定:=ILI (幅データ入力, 高さデータ入力)
幅データ入力:=入力選択 >> 入力*
      UA:mouse-click"入力エリア" UA:key-press" * "
      [> 設定
      UA:mouse-click"→入力エリア"
高さデータ入力:=入力選択 >> 入力*
      UA:mouse-click"入力エリア" UA:key-press" * "
      [> 設定
      UA:mouse-click"→入力エリア"

```

図 3:TID 仕様例

そこで本稿では TID 仕様での、インタラクション構造における一貫性に注目し、その一貫性を分析・評価するための手法を提案する。

3.1 分析手法 (概念)

「同じ様な時に、同じ様な事を、同じ様に出来る」という UI システムの一貫性は、ユーザの理解や学習を促進すると考えられる。つまり、こうした一貫性を兼ね備える UI システムにおいてユーザは、一度獲得した操作法から他の機能に対する操作法を容易に予測することが可能となり、またそれにより効率的にシステム全体の操作法を容易に学習することが可能になるという効果が期待出来る。そこで、本手法では「同じ様な時に、同じ様な事」をしているタスク、つまりそれぞれのタスクが持つ意味的な類似性を、そのタスクが持つインタラクション構造から判別し、それに対し「同じ様に出来る」かどうかを具体的なユーザの操作ユーザアクション(UA)として考える。

「TID 仕様において同じインタラクション構造を持つタスクは意味的な類似性を持つ

ている」という仮定に基づき、インタラクション構造とユーザアクション(UA)とのマッピングにおける規則性をインタラクションの一貫性として捉え、そうした規則性を逸脱している可能性のあるタスクを抽出する手法が本稿の提案する分析法となる。

同じインタラクション構造を持つタスクは、高い確立で意味的な類似性を持つことが実験により確認されており、例えば、ある既存の PC 上で動作するアプリケーションでは、その TID 仕様 (UI 機能 47 個、関係子 79 個) から、同じインタラクション構造を持つタスクを抽出したところ平均で約 85%のタスクが意味的な類似性を持つタスクであった。

3.2 分析手順

分析手順を図 4 に示す。

前提

分析者は全ての UI 機能の内容を把握しており、類似した機能を提供する UI 機能を UI 機能群として類別した情報を持っている。(もしくは設計者から与えられる。)

STEP1

分析者は、どのようなインタラクション関係を持つタスクを分析対象とするのかを決定する。ここで分析者は以下の 7 つのセットで与えられたインタラクション関係のうち一つを選択し、そのインタラクション関係を持つタスクを全て TID 仕様から抽出する。

(1) **>>***

任意の二つのタスクが sequence の関係にあるタスク。右側のタスクは、そのタスクで詳細化されている被定義タスクの終端タスクではない。

(2) **>>***

任意の二つのタスクが sequence の関係に

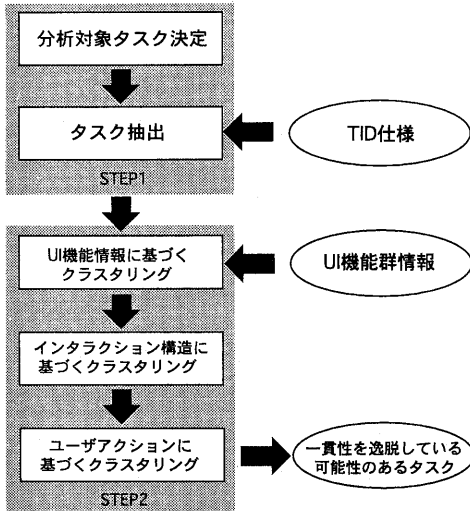


図 4:分析の流れ

あるタスクかつ、右側のタスクが、そのタスクで詳細化されている被定義タスクの終端タスクである。

(3) ** [>]**

任意の二つのタスクが interrupt の関係にあるタスク。右側のタスクは、そのタスクで詳細化されている被定義タスクの終端タスクではない。

(4) ** [>]**

任意の二つのタスクが interrupt の関係にあるタスクかつ、右側のタスクが、そのタスクで詳細化されている被定義タスクの終端タスクである。

(5) ** [>>]**

任意の二つのタスクが skippable の関係にあるタスク。右側のタスクは、そのタスクで詳細化されている被定義タスクの終端タスクではない。

(6) ** [>>]**

任意の二つのタスクが skippable の関係にあるタスクかつ、右側のタスクが、そのタスクで詳細化されている被定義タスクの終端タスクである。

(7) | (**)

タスクが choice の関係を持つタスク。

STEP2

STEP1 で抽出されたタスクに対し、以下に示す3つのクラスタリングを行う。

■ UI 機能情報に基づくクラスタリング

分析対象のタスクが持つ UI 機能情報に注目したクラスタリング。前述の UI 機能群ごとに、含まれる UI 機能を UI 機能情報として持つタスクを同じクラスタに分類。

■ インタラクション構造に基づくクラスタリング

分析対象のタスクが詳細化する被定義タスク（親タスク）の持つタスク関係に注目したクラスタリング。親タスクの関係子が同じタスクを同じクラスタに分類。

■ ユーザアクション(UA)に基づくクラスタリング

分析対象のタスクに割り当てられているユーザアクションに注目したクラスタリング。同じUAを持つタスクを同じクラスタに分類。

各クラスタリングは、次に示す (α), (β) いずれかの順序で実行する。

(α) インタラクション構造に基づくクラスタリングを実行後、それぞれのクラスタごとにユーザアクションに基づくクラスタ

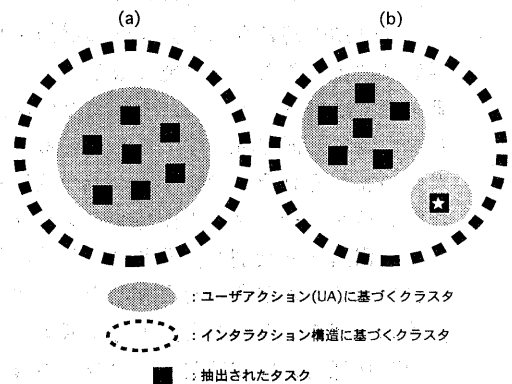


図 5:典型的なクラスタ

抽出対象タスク：**[>]** 親タスクの関係：l(choice)		
タスク番号	抽出タスク	ユーザアクション(UA)
1	ファイル選択 [> 決定	(なし) [> click-mosue"ボタン"
2	保存設定 [> 決定	(なし) [> click-mouse"ボタン"
3	各パラメータ設定 [> 描画決定	(なし) [> click-mouse"ボタン"
4	各パラメータ設定 [> 描画決定	(なし) [> click-mouse"ボタン"
5	頂点決定* [> 終点決定	key-press"*"* [> click-mouse"→入力エリア"
6	入力* [> 決定	key-press"*"* [> click-mouse"ボタン"

図 7:インタラクション構造に基づくクラスタ(1)

抽出対象タスク：**[>]** 親タスクの関係：l(l(parallel last)		
タスク番号	抽出タスク	ユーザアクション(UA)
7	入力* [> 決定	key-press"*"* [> click-mouse"→入力エリア"
8	設定* [> 決定	key-press"*"* [> click-mouse"→入力エリア"
9	頂点決定* [> 終点決定	key-press"*"* [> click-mouse"→入力エリア"
10	入力* [> 決定	key-press"*"* [> click-mouse"→入力エリア"

図 8:インタラクション構造に基づくクラスタ(2)

ングを実行する。

(β)UI 機能情報に基づくクラスタリングを実行後、それぞれのクラスタごとに(α)を実行する。

3.3 評価手順

前記 3 種の異なるクラスタリングで作成されたクラスタに含まれるタスクの差異を分析することで一貫性を逸脱している可能性のあるタスクを発見、抽出する。「同じ様な時に同じ様な事」をしているタスク、すなわち意味的な類似性を持つタスクに対して、ユーザは同じ操作で実行可能であることが望ましい。つまり「インタラクション構造に基づくクラスタ」に含まれるタスクを全て含む「UA に基づくクラスタ」が一つのみ存在することが、この場合の望ましい「一貫性」と言える。例えば図 5(a)のように「あるインタラクション構造に基づくクラスタ」に含まれるタスクが一つの「UA に基づくクラスタ」を形成している場合、これらのタスクの中に上記の一貫性

を逸脱している可能性のあるタスクは存在しないと考える。図 5(b)のように、ある「インタラクション構造に基づくクラスタ」に含まれるタスクが複数の「UA に基づくクラスタ」を形成し、かつほとんどのタスクが一つの大きなクラスタを形成仕手入る場合、極端に少ないタスクを含むクラスタのタスク(★印のタスク)は上記の一完成を逸脱している可能性のあるタスクとして考えられる。

(α)の手順でクラスタリングを行った場合は、分析対象のアプリケーション全体の一貫性を分析することになり、アプリケーション全体で保たれているべき普遍性を分

描画関係UI機能群
方形描画, 円形描画, 多角形描画, 直線描画, テキスト入力, オブジェクト選択, カラーペイント
ファイル操作関係UI機能群
新規作成, 開く, 閉じる, 保存

図 6:UI 機能群

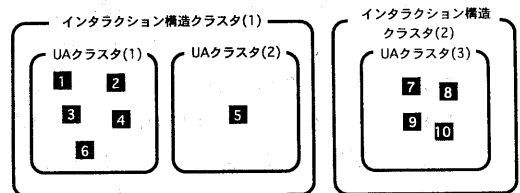


図 9:UA に基づくクラスタリング(1)~(3)

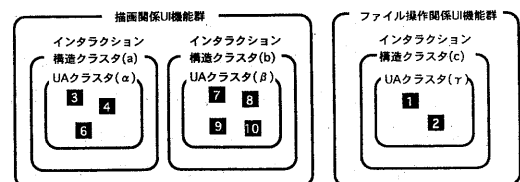


図 10:UA に基づくクラスタリング(α)~(γ)

析/評価することになる。(β)の手順でクラスタリングを行った場合は、タスクを機能的な側面で分割して考えた場合の一貫性を分析/評価することになる。つまり、(α)の結果と(β)の結果を比較することにより、アプリケーション全体で保たれているべき一貫性、機能ごとに異なるべきである一貫性とを区別して分析することが可能になる。

3.3 分析例

11個のUI機能が定義された架空のペイント・アプリケーションのTID仕様に対する分析例を示す。UI機能群を図6のように定義し、分析対象のタスクが持つインタラクション関係を「**[>]**」として分析した場合を考える。

「インタラクション構造に基づくクラスタ(インタラクション構造クラスタ)」に含まれるタスクを図7、図8に示し、それぞれのクラスタに含まれるタスクに対してUAに基づくクラスタリングを施した結果((α)の手順でクラスタリングを行った場合)を図9に示す。図8のインタラクション構造クラスタ(2)に含まれるタスク7-10は全て、一つの「UAに基づくクラスタ(UAクラスタ)」を形成しているため一貫性を逸脱しているタスクは存在しないと考えられるが、図7のインタラクション構造クラスタ(1)に含まれるタスクにおいては、タスク5のみ異なるUAクラスタ(2)として分類されているため、タスク5を「一貫性を逸脱している可能性のあるタスク」として抽出することが出来る。実際の仕様において、タスク1-6は全てテキスト入力ボックスを持つダイログ・ウインドウに対する決定操作を意味するタスクであるが、タスク5のみが異なる操作を割り当てられていることになる。「同じ様な時に同じ様な事」をするタスクの中に一つ、このような例外が存在するために、ユーザは混乱を生じる可

能性が存在するため、抽出されたタスク5は再検討を要するタスクと考えることができる。

更に(β)の手順で行ったクラスタリングの結果を図10に示す。図9では同じUAクラスタ(1)に含まれていたタスク1, 2, 3, 4, 6がそれぞれ異なるUI機能クラスタとしてクラスタリングされている。つまり、タスク3, 4, 6とタスク1, 2は、機能的な側面で捉えた場合、それぞれ異なる状況で遂行されるタスクなのにも関わらず同じ操作が割り当てられていることになる。こうした一貫性は逆にユーザの混乱を生じる可能性もあり、注意が必要なタスクとして考えるべきである。

4. おわりに

本稿ではTID仕様に対するインタラクション構造とユーザアクションのマッピング規則に注目したインタラクションの一貫性を分析する手法を提案した。本手法による分析・評価結果をTIDによる設計仕様をよりユーザビリティの高いものへと洗練するためのフィードバックに成り得ると考えられ、より効率的なUI設計が期待できる。

今後は、本分析手法を支援するための分析支援ツールの設計・実装を目指し検討を進めると共に、本分析手法の有用性を評価するための更なる実験を行う予定である。

参考文献

- [1]石山 啓太郎, 木下 哲男, 白鳥 則郎「インタラクション設計記法 TID に基づくユーザインターフェース設計仕様分析手法の検討」SSE98-89~107, pp.7-12, 電子情報通信学会(1998)
- [2]Y. Tokuda, E. S. Lee, N. Shirator「Synthetic and Analytic Method for User-Computer Interaction Design」Proceedings the 12th Interaction Conference on Information Networking, pp.726-729(1988)