

タスク指向ユーザインタフェース アーキテクチャの設計支援環境の検討

伊澤 謙一

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1-1-1

TEL:022-217-5454

E-mail:{isawa,kino,norio}@shiratori.riec.tohoku.ac.jp.

近年のユーザインタフェース(UI)に対する要求の多様化に対し、そのようなユーザ要求を十分反映したUIを効果的に設計することの必要性が高まっている。本稿は、著者らが提案しているUIアーキテクチャ設計記法 TIO(Task oriented Interaction Object)に基づいたUIアーキテクチャ設計支援手法を提案する。この設計支援手法は、TIO仕様のリポジトリによる獲得支援と、ルールベースによるTIO制御構造の定義支援を行う。更に本稿では、この設計支援手法を実装したUIアーキテクチャ設計支援環境を紹介について述べる。

Support Environment for Task Oriented User Interface Architecture Design

Ken'ichi Isawa

Tetsuo Kinoshita

Norio Shiratori

Research Institute of Electrical Communication / Graduate School of Information Sciences,
Tohoku Univ.

2-1-1, Katahira, Aoba-ku, Sendai 980-8577

TEL:+81-022-217-5454

E-mail:{isawa,kino,norio}@shiratori.riec.tohoku.ac.jp

With the diversity of requirements of user interfaces(UI), it is essential that effective UI design which can reflect various user requirements. In this paper, we propose a method to support UI architecture design based on TIO (Task oriented Interaction Object) method proposed by authors. This method-based supports designers to define TIO specifications using TIO repository and describe TIO structure based on rule-based mechanism. And then, we discuss a support environment of UI architecture design based on the proposed method.

1 はじめに

近年のユーザインタフェース(UI)に対する要求の多様化に伴い、ユーザ要求を十分に反映したUIを効果的に設計することの必要性が高まっている。これに応えるため、ユーザとシステムの対話、即ちインタラクションをタスクという観点で捉えて設計するタスク指向インタラクション設計が考案されている[1]。

著者らはUIアーキテクチャモデルTIO(Task oriented Interaction Object)[2]とそれに基づいたUIアーキテクチャ設計支援手法[3]を提案している。ここではUI設計の流れを図1のように捉えている。まず最初に設計者は、インタラクション設計環境においてユーザとUIシステムの間のインタラクションを設計し、インタラクション仕様を獲得する。設計内容の記述には、著者らが提案しているタスク指向インタラクション記述法TID(Task oriented Interaction Description technique)が用いられる。

設計者は次に、獲得されたインタラクション仕様通りに動作するUIアーキテクチャを設計する。これは、UIを構成するオブジェクトの仕様を具体化し、そしてオブジェクト間の制御関係を定義することにより行われる。UIアーキテクチャの仕様を表現するためのモデルがTIOであり、本稿で提案する設計支援環境がその仕様設計の支援を行う。

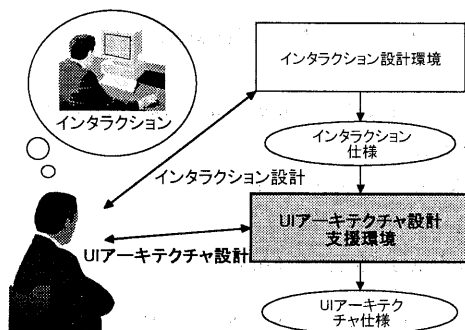


図1: UI設計のプロセス

2 UIアーキテクチャモデルTIO

2.1 TIOの内部アーキテクチャ

TIOはUIアーキテクチャを構成するコンポーネントである。UIアーキテクチャ全体は、TIOを階層的に構造化することによって獲得される。

TIOの基本的なアーキテクチャを模式的に示したのが図2である。TIOは6つのゲートを持ち、TIO間でこのゲートを介してデータやイベントのやりとりを行うことができる。基本的に以下の4通りの動作を行うことができる。

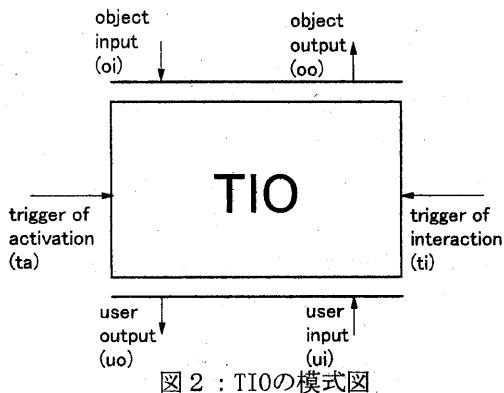
1. ユーザからの入力を用user input(ui)のゲートで受け取る。また、UIオブジェクトとしての外見や、ユーザの操作に対する視覚的なフィードバックを用user output(uo)から出力する。
2. 他のTIOからのデータの用object input (oi)で受け取る。また、他のTIOへのデータの出力を用object output (oo)から送り出す。
3. インタラクションの区切りを知らせるイベントを用trigger for interaction (ti)で受け取る。TIOはそれまでのインタラクションで蓄積したデータをooから出力する。
4. TIOの動作を制御するためのイベントを用trigger for activation (ta)で受け取る。TIOは受け取ったイベントに従って自らの動作を制御する。

各ゲートの動作と処理するデータの形式を定義することが、即ちTIOの仕様設計になる。

2.2 TIOの特徴

TIOの特徴として以下のような点が挙げられる。

1. TIDとの親和性をはかっているため、インタラクション仕様からアーキテクチャ仕様を導出することが容易になっている。
2. 動作に関する記述を内部に持つので、具体的なUIのプログラムコードを導出することが容易になっている。

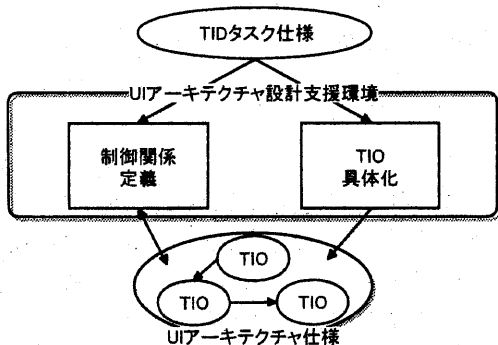


3 UIアーキテクチャ設計支援環境

ここではUIアーキテクチャ設計支援環境について述べる。まず最初にUIアーキテクチャ設計の全体的な流れについて述べ、次に著者らが提案している設計手法を解説する。そして最後に、この設計手法に基づいた設計支援環境を紹介する。

3.1 UIアーキテクチャ設計プロセス

本稿では、UIアーキテクチャ設計の流れを図3のように捉えている。



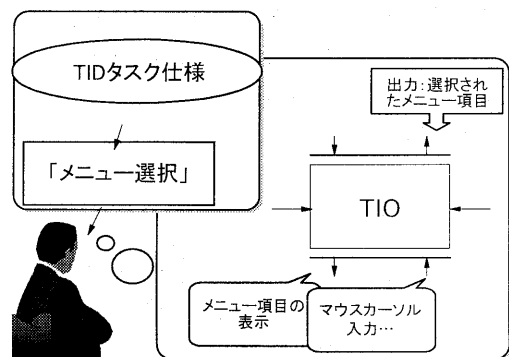
UIアーキテクチャ設計は、次の2つのプロセスから構成されている。

1. TIO具体化
2. 制御関係定義

3.1.1 TIO具体化

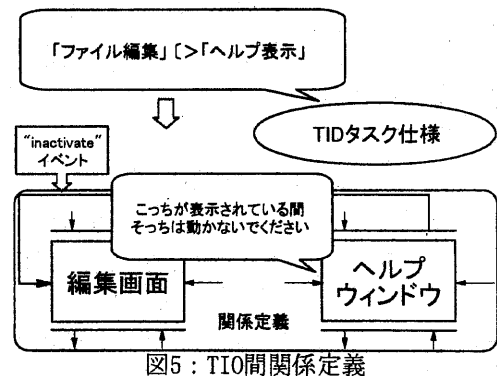
UIオブジェクト具体化では、入力として与えられたタスクを実際に行うUIオブジェクトの仕様をTIOで記述する。例えば、タスクとして「メニュー選択」というものが与えられたならば、プルダウンメニューやメニューバーというUIオブジェクトをTIOで記述するのがこのプロセスになる。具体的に定義する項目としては、

- UIオブジェクトとしての外見
- 扱うデータの内容
- …などが挙げられる。



3.1.2 TIO間関係定義

TIO間関係定義では、具体化されたTIOの間の時間的な制約条件や制御を定義する。具体的には、TIOの間でやりとりされる制御イベントの定義がこれにあたる。



この設計を行うには、TIOタスク仕様に記述されているタスク間の時間的關係についての情報が必要になる。

3.2 各プロセスにおける設計支援手法

3.2.1 TIO具体化支援手法

TIOの具体化を容易にするために、支援手法としてTIOのリポジトリを用意する。このリポジトリには、さまざまなTIOの設計仕様が蓄積されている。設計者は、与えられたインタラクション仕様を実現するのに最適なTIOをリポジトリから獲得し、再利用することができる。こういう仕組みを用意することによって、設計者の設計の手間を軽減することを狙いとしている。

このリポジトリを利用した設計の流れを図4に示す。

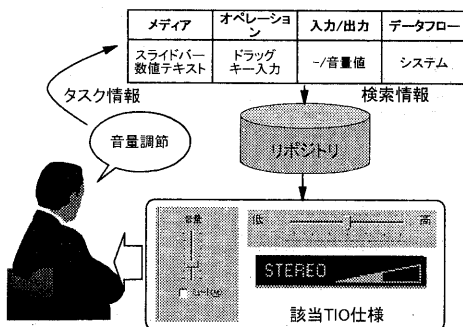


図6: リポジトリを用いたTIOの具体化

リポジトリに蓄積されているTIOは、以下に示す項目に従って分類され、インデクシングされる。

1. メディア…TIOが視覚的にどのような外見を持っているか。
2. オペレーション…ユーザはTIOに対してどのような操作をすることができるか。
3. 入力/出力…インタラクションの結果、TIOはユーザからどのようなデータを受け取るか、そして、外部にどのようなデータを出力するか。
4. データフロー…出力データはどこに対して送られるか。

設計者は、具体化しようとしているタスク仕様を

もとに検索情報を作り出し、リポジトリに対して与えてやる。例えば図4の場合、「音量調節」というタスクが要求するTIOは、メディアとしてスライダーや数値テキストを持ち、ドラッグや数値を入力するというオペレーションが可能で、音量を出力データとして扱うだろうということを設計者が分析する。そしてそれらの項目をキーとしてリポジトリに検索情報として与えることで、目指すTIO仕様を獲得することができることになる。

3.2.2 TIO間関係定義支援手法

TIO間関係の定義をするには、TIO間でやりとりされるイベントの記述などが必要になるが、この作業を自動化することが本支援手法の狙いである。

具体的には、入力として与えられるタスク仕様記述からTIO間関係を自動的に導出するルールベースを用意する。

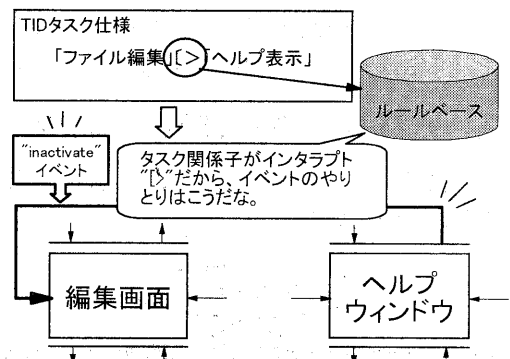


図7: ルールベースを用いたTIO間関係定義

入力として、既に具体化されたTIOと、それらに対応するタスクの仕様を記述したTIDタスク仕様を与える。ルールベースはこれらの入力を解析して、TIO間の適切な制御関係を定義する。

3.3 各設計支援手法の実装モジュール

3.3.1 TIO具体化支援モジュール

3.2.1で解説したTIO具体化支援手法は、以下に示す構成によって実装される。大きく3つの機

能から成り立っており、それぞれの概要は以下の通りである：

1. 検索情報抽出…入力として与えられたタスク仕様から、TIO仕様を検索するために必要な情報である情報検索を作り出す。
2. TIO仕様検索…検索情報をもとに、リポジトリからTIO仕様を検索する。検索情報に適合するTIO仕様の候補がTIO仕様群として抜き出される。
3. 採用…検索の結果抽出されたTIO仕様群の

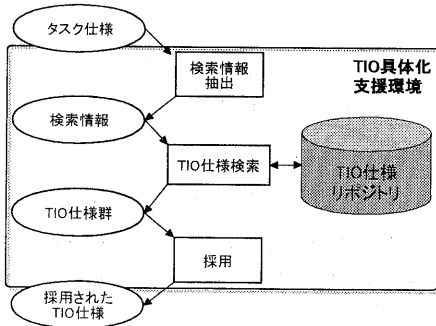


図8:TIO具体化支援の機能構成

中から、設計者がもっとも適切だと考えるTIOを選択し、設計仕様として採用する。

この構成に基づいて実装された、TIO具体化支援環境のインタフェースプロトタイプを図9に示す。

3.3.2 TIO間関係定義支援モジュール

3.2.2で解説したTIO間関係定義支援手法は、以下に示す構成によって実装される。

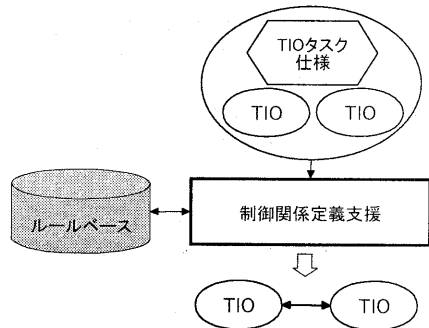


図10:TIO間関係定義支援の機能構成

4.本提案手法を用いた設計例

本設計支援環境を利用した簡単な設計事例を紹介する。

まず図11に示すTIDタスク要求が支援環境に

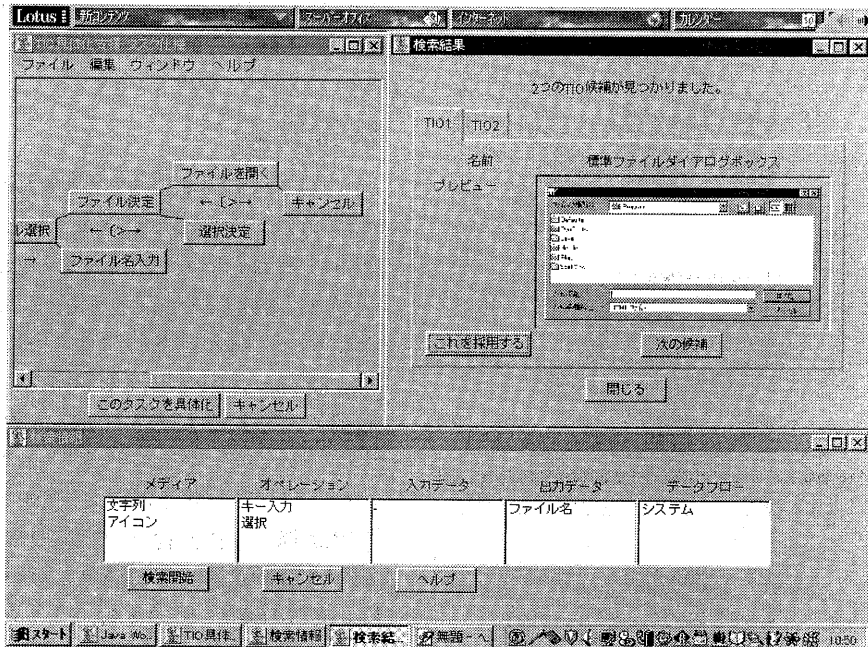


図9:TIO具体化支援環境のユーザインタフェース

与えられたとする。このTIDタスク仕様に記述されているのは、ファイルを開くというタスクの内容であ

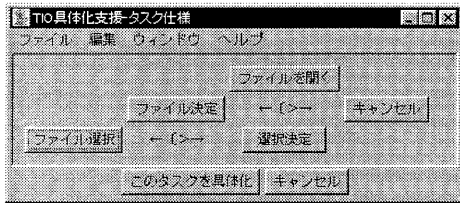


図11：TIDタスク仕様

る。最初に、「ファイル選択」というタスクを実行するTIOの仕様を具体化する。図12がその様子を示したものである。

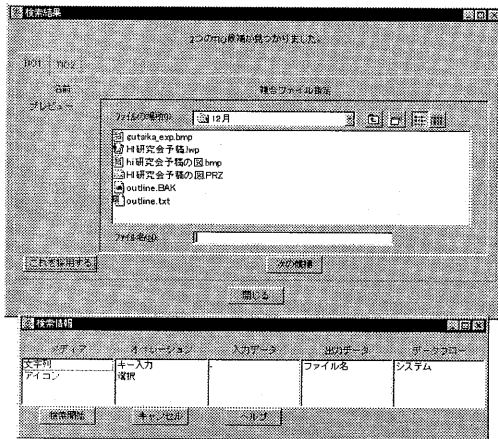


図12：「ファイル選択」のTIOの具体化

次に、「選択決定」のタスクを実行するTIOの仕様を具体化する。図13がその様子を示したものである。

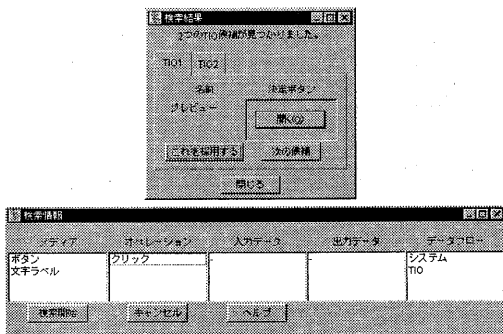


図13：「選択決定」のTIOの具体化

次に、この2つのTIO間関係を定義する。この様

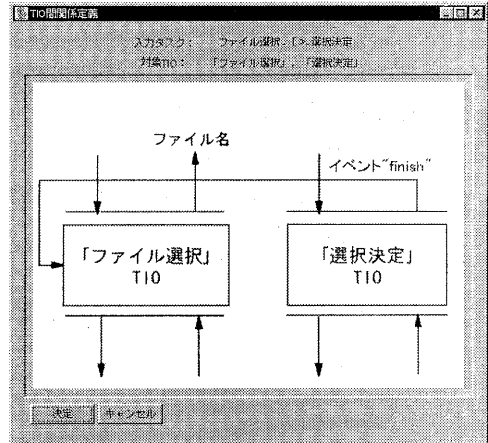


図14：TIO間関係の定義

子を図14に示す。

以下同様にして、他のタスクについてもTIOを具体化し、関係づけをしていく。これを繰り返すことで最終的にUIアーキテクチャ設計が完了する。

5 おわりに

本稿は、タスク指向UIアーキテクチャ設計記法TIOに基づいたUIアーキテクチャ設計支援環境とその手法を提案した。今後は本設計支援環境の実装を完了し、評価を行う予定である。

参考文献

- [1] Y. Tokuda, E. S. Lee and N. Shiratori: Synthetic and Analytic Methods for User-Computer Interaction Design, Proc. of the 12th International Conference on Information Networking, IEEE Computer Society, pp.726-729, (1998).
- [2] 伊澤 謙一, 徳田 佳一, 村松 成治, 田中 康, 李 殷碩, 白鳥 則郎: タスク指向インタラクション設計に基づくUIアーキテクチャ設計法の検討, 情処研報, HI77-5, 情報処理学会(1998).
- [3] 伊澤 謙一, 木下 哲男, 白鳥 則郎: UIアーキテクチャ設計記法TIOに基づくUIアーキテクチャ設計支援の検討, 信学技報, IN98-70, 電子情報通信学会(1998).