1N-02

GUI 設計のためのモデルベースデザイン評価手法

山田 龍平† 鷲崎 弘宜† 深澤 良彰† 鹿糠 秀行‡
早稲田大学大学院 基幹理工学研究科 情報理工・情報通信専攻†
株式会社 日立製作所 研究開発グループ‡

1. はじめに

GUI アプリケーションのユーザビリティを評価することを目的に、画面デザインが既知のデザインパターンに適合しているかどうかをソースコードで確認する方法がある。しかし、この方法では言語ごとに文法が異なるため、プログラミング言語に依存する問題がある。

そこで、画面デザインをプログラミング言語に依存しないモデルとして定義し、モデルを通じて既知のデザインパターンの適合の有無を確認するモデルベース画面デザイン評価手法を提案する。

本論文では、画面デザインを表現するために KDM をベースとしたメタモデルと、既存のいくつかのデザインパターンの適用有無を確認するための OCL 実装例を示し、提案手法の妥当性を議論する。

2. モデルベースデザイン評価手法

ソースコードではなくプログラミング言語に依存しないモデルに対してデザインパターンを検出することで、GUI アプリケーションの画面デザインを評価するモデルベースデザイン評価手法を提案する。

2.1. 画面デザインのためのメタモデル

メタモデルにはソフトウェアの様々な要素や関連を表現できる KDM (Knowledge Discovery Metamodel)[1]を採用する。その中で基本的な画面デザインの要素と構成を表現する UIPackage をベースに、携帯端末などの画面を表現できるように拡張し、またデザインパターンを検出する手がかりとなる要素を追加する。

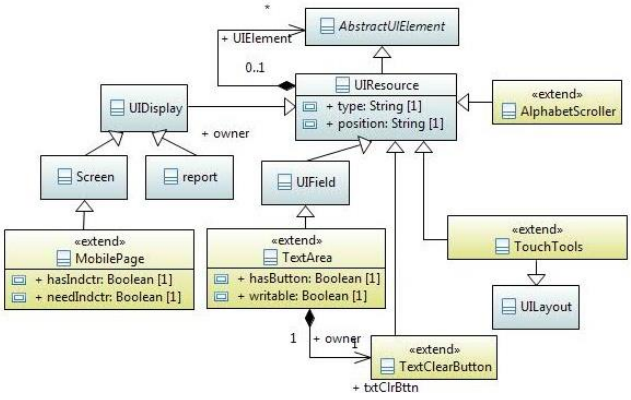

図 1 Extended KDM

KDM メタモデルを拡張して定義したメタモデル(Extended KDM)の一部を図 1 に示し、拡張した一部のクラス(ステレオタイプ<<extend>>を記入したクラス)について以下にて説明する。

・**TextArea**：画面上の任意のフィールドを示す UIField を拡張して、文字が書込可能なフィールドまたは文字表示だけのフィールドを表現するテキストエリアを表現する。

・**MobilePage**:画面を表現する Screen を拡張し、携帯端末画面を表現する。

・**TextClearButton**：テキストを一括消去するボタンを表現する。

2.2. モデルに対するデザインパターン検出方法

画面デザインにデザインパターンが適用されているかどうかをモデル上で検出するために、モデルの制約記述言語である OCL (Object Constraint Language)を利用する。あるデザインパターンが満たすべき条件をモデル上の制約として定義し、これを OCL で実装する。

文献[2]で提案されている携帯端末に関する画面デザインのデザインパターンの中のいくつかを題材に、Extended KDM に基づくモデル上でそれらデザインパターンを検出するための OCL での実装例を表 1 に示す。

表 1 画面デザインのデザインパターンと OCL 実装例

デザインパターン	OCLでの実装例
サムネイルとテキストのリスト	context UIResource inv checkText: self.type = 'Text' implies self.position = 'RIGHT' inv checkThumb: self.type = 'Thumbnail' implies self.position = 'LEFT'
タッチ表示ツール	context TouchTools inv ToolSwitching: self.toolSwitch = true and (self.to.type = 'MotionPicture' and self.from.type = 'Tools')
テキスト消去ボタン	context TextArea inv checkButton: self.writable = true and self.hasButton = true context UILayout inv buttonIncluding: self.to.type = 'TextArea' and self.from.type = 'TextClearButton'
縦方向のスタック	context UIResource inv checkStack: self.type = 'Contents' implies self.position = 'CENTER'
無限リスト	context UILayout inv checkInfinity: self.from.type = 'InfinityButton' and self.to.type = 'Window' context UIResource inv checkPosition: self.type = 'InfinityButton' implies self.position = 'UNDER'

The Method of Model-Based Design Evaluation for GUI Designing
†Ryuhei Yamada †Hironori Washizaki
†Yoshiaki Fukazawa ‡Hideyuki Kanuka
†Waseda University ‡Hitachi, Ltd.

3. モデルベースデザイン評価の手順

Extended KDM と OCL を利用したモデルベースデザイン評価の手順を例で示す。ここで想定する画面イメージを図 2 に示す。なお、この画面イメージは携帯端末対応のブラウザアプリの想定である。

この画面デザインには、テキスト消去ボタンパターンと縦方向のスタックパターンの 2 つが適用されている。

・テキスト消去ボタンパターン：ボタンを 1 回タップする操作によって、文字書込可能なフィールド内のテキストを消去できるようにする。

・縦方向のスタックパターン：モバイル向けページのコンテンツを縦 1 カラムの構成で示し、横並びの要素を用いないようにする。

3.1. 画面デザインのモデル化

図 2 に示した画面デザインを、Extended KDM のモデルで表現した様子を図 3 のオブジェクト図に示す。

画面を表す Screen は、TextClearButton を持つ TextArea と、2 つのコンテンツ(contents1 と contents2)を表現する UIResource を持っている。

TextArea は属性として文字の書込の有無を表現する writable を持ち、ここでは true である。

図 2　画面デザイン例

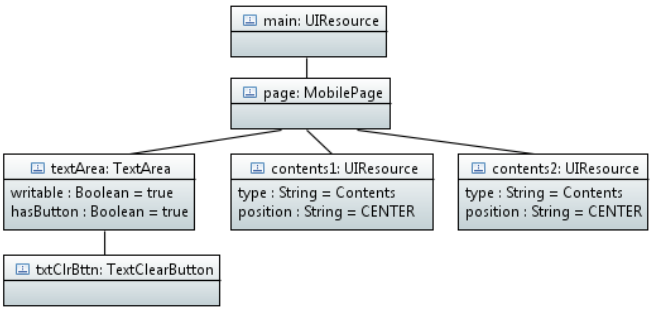

図 3 画面デザインのモデル表現

UIResource は属性として type を持ち自身がコンテンツ(contents)であること表し、位置を表現する position を持つ。ここでは contents1 と contents2 が共に CENTER(中央)であり同列に並んでいることを表現している。

3.2. OCL によるデザインパターンの検出

ExtendedKDM のモデルに対して OCL 実装(表 1)を適用することで、モデルに適用されているデザインパターンを検出する。

本研究での OCL の実行は、Eclipse OCL[3]を利用し、図 2 に適用されている 2 つのパターンを図 3 のモデル上で検出することができた。

テキスト消去ボタンパターンの OCL 実装では、TextArea に対して writable 属性が true かつ TextClearButton を持つことを、制約を通じてチェックでき当該パターンを検出できる。

また縦方向のスタックパターンの OCL 実装では、type が contents である場合に、position が CENTER であることを制約を通じてチェックでき当該パターンを検出できる。

なお制約を満たさない場合は、Eclipse OCL によって制約違反が出力されるので、画面デザインが既知のデザインパターンに適合していないことを検出できる。

4. おわりに

GUI アプリケーションの画面デザインをモデルを通じて評価するモデルベースデザイン評価手法を提案した。いくつかのデザインパターンが画面デザインに適用されている場合に、実際それらを検出できることを明らかとした。

今後は検出できるデザインパターンの拡充を考えている。また画面からモデルを作る作業は現在人手のため、画面を実装したソースコード等からモデルを自動抽出する技術を開発・組合せることで効率化を図り、GUI アプリケーションのユーザビリティ評価手法を発展させていく。

参考文献

[1] OMG: Architecture-Driven Modernization: Knowledge Discovery Meta-Model (KDM) Version 1.4, www.omg.org/spec/KDM/1.4/PDF

[2] Jenifer Tidwell、ソシオメディア株式会社（監修）、浅野紀子（訳）：デザイニング・インターフェース 第 2 版 —パターンによる実践的インタラクションデザイン、オライリージャパン（2011）