

平成26年度
信州大学工学部情報工学科卒業論文

画面遷移モデルを基にしたシナリオ 作成支援

海尻・小形研究室
学籍番号 11T5048H

中村 哲真

目 次

1	序論	3
1.1	背景	3
1.2	目的	3
1.3	論文の構成	3
2	用語説明	4
2.1	画面遷移モデル	4
2.2	画面インスタンス	4
2.3	シナリオ	5
3	関連研究	5
4	提案手法	5
4.1	画面遷移モデルの解釈	6
4.2	シナリオ作成に必要な要素	7
4.3	シナリオの導出	7
5	評価	7
5.1	評価方法	7
5.2	考察・結果	7
6	結論	7
6.1	まとめ	7
6.2	今後の課題	7

1 序論

1.1 背景

ソフトウェア開発の上流工程では、主に設計が行われる。この時、適切に設計されていることは重要である。なぜなら、設計はソフトウェア開発全体の基となり、設計に含まれる問題は実際のソフトウェアに含まれてしまうためである。また、設計に問題がある場合、早期にそれを発見することも重要である。問題を解決する際には、問題にかかわる実装済みの部分をすべて修正する必要がある、発見が遅れると修正箇所が多くなるためである。

設計段階に問題を検証する方法として、入力保存機能に着目したモデル駆動ユーザビリティ評価法の提案と評価 [pending] が提案されている。この手法では UML (Unified Modeling Language) を拡張し、システムの画面遷移モデルと、システムの利用上想定されるシナリオを定義している。これにより、上流工程でユーザビリティに関する問題が発見できる。しかし、この手法のシナリオを手動で入力することは煩雑である。例えば、この手法ではシナリオを作成する時、多くの類似した値を入力しなければならない。また、正しいシナリオを作成するには手法を熟知している必要がある。この手法では UML を拡張しているため、拡張部分を理解しなければ正しいシナリオを作成できない。

このように手法の利用に対する障壁が高い場合、実際の開発では適用されない可能性が高くなる。従って、実用するためには利用に対する障壁を取り除く必要がある。

1.2 目的

本研究は、ソフトウェア開発の上流工程において、特にユーザビリティに関する問題の発見を支援することを目的とする。上流工程でユーザビリティの評価を行う手法は存在する¹が、利用に対する障壁が大きい。

上流工程で発生した問題を早期に発見するために、入力保存機能に着目したモデル駆動ユーザビリティ評価法の提案と評価 (以下、「適用する手法」)¹で提案されているシナリオの作成支援を行う。

1.3 論文の構成

第二章では適用する手法に関する語句を説明する。第三章では関連研究について述べる。第四章では手法の説明を行い、第五章では評価について述べる。第六章ではまとめと今後の課題について述べる。

¹入力保存機能に着目したモデル駆動ユーザビリティ評価法の提案と評価

2 用語説明

2.1 画面遷移モデル

画面遷移モデルとは、システム上の画面間の遷移を定義するモデルである。クラス図上に画面および遷移が記述される。画面遷移モデルの記法はメタモデルで定義されている (図??)。

記法について説明する。画面 (Screen) はクラスとして表され、名前 (Screen.name) および操作項目 (OperatableItem) を持つ。操作項目は名前およびユーザー意図 (OperatableItem.ui) を持ち、遷移 (Transition) と入力項目 (InputItem) に分かれる。遷移は対象の画面を持つ。入力項目には入力型 (InputType) があり、入力された値の保存範囲を持つ。また、平均入力量 (metricsOfUserEffort.avg) と一画面入力限度数 (metricsOfUserEffort.upto) を持つ。入力項目は文字列入力 (InputText)、単一選択 (SingleSelection)、複数選択 (MultipleSelection) に分かれる。

ユーザー意図はユーザーが画面遷移する動機を表し、作成 (Create)、閲覧 (Browse)、更新 (Update)、削除 (Delete)、取消 (Cancel) の5種類に分けられる。また、ユーザー意図は操作対象画面 (UserIntention.ots) を持つ。これはユーザー意図の対象となる画面を表し、操作対象が遷移先でない場合に明示される。

2.2 画面インスタンス

画面インスタンスとは、画面 (Screen) のインスタンスで、各画面で入力された値が格納される。これには以下の6個の拡張記法が定義されている。

<<delete>> 画面インスタンスが削除されたことを示す。

<<update>> 画面インスタンスが更新されたことを示す。

<<cancel>> 入力された値が取り消されたことを示す。

<<destination>> 意図の中で最後に到達する画面インスタンスであることを示す。

\a 入力値および画面インスタンス名に対する拡張記法。入力値がシステムによる自動入力であることを示す。

\, 入力値に対する拡張記法。一つの項目に対して複数の値を入力する際に用いる。

2.3 シナリオ

適用する手法のシナリオは、想定されるユーザーの入力が記述される。また、画面遷移モデルが変更された際にシナリオを修正する必要があるため、シナリオに操作順序は記述されない。

シナリオは意図と呼ばれる単位で分割される。意図とはユーザーが行動する指向のことである。例えば、商品を購入する、カートに入れた商品を修正する等である。

シナリオは複数のオブジェクト図で記述され、ユーザーの入力は画面インスタンスで記述される。オブジェクト図は以下の 5 通りに分類可能で、シナリオには全て必要がある。

開始 (Start) シナリオの起点となる画面インスタンスを一つ含む

事前条件 (Precondition) シナリオ開始前に存在する画面インスタンスを任意の数含む

意図 (Intention) 各意図で入力される画面インスタンスを含む

事後条件 (Postcondition) シナリオ完了後に存在している画面インスタンスを含む

終了 (End) シナリオの終点となる画面インスタンスを一つ含む

3 関連研究

4 提案手法

本研究では、シナリオの作成支援ツール (以下、「本支援ツール」) を実装した。プラグインにより拡張可能であり、適用する手法が利用していることから、本支援ツールは astah* を利用した。また、astah* Plug-in は Java で記述する必要があるため、実装は Java で行い、GUI 部品の表示は swing, awt を利用した。

本支援ツールは画面遷移モデルを解釈して対象のシステムをシミュレートし、開発者が実際に操作してシナリオを作成する。すなわち、web ブラウザーのように画面遷移モデルをブラウジングし、値を入力することでシナリオを作成する。これにより、以下の効果を期待する。第一に、拡張記法をツールが補完し、手法の初学者が文法的な誤りを犯さない効果。第二に、ツールが以前入力した値を保持し、入力量を低減する効果である。

図??に本支援ツールの構造を示し、主要なクラスを解説する。

SupportSystem はエントリーポイントである。このクラスはシナリオ作成全体の状態を持ち、Browser, ScenarioCreator, ModelListener を初期化する。また、GUI を実際に描画する部分も担う。

Browser は核となる部分である。このクラスは Screen（画面）の表示と画面遷移を行う。Screen の表示は、GUI 部品の構成を担う ComponentManager に Screen を渡し、それを表示する。画面遷移は、ButtonComponent から通知を受け遷移を行う。また、History への値の格納も行う。

ScenarioCreator は、History に格納された値および、設定された Precondition（事前条件）を元にシナリオを生成する。

4.1 画面遷移モデルの解釈

対象のシステムをシミュレートするため、画面遷移モデルを解釈し、Screen（画面）から GUI を構成する。Screen は OperatableItem を保持しており、これは GUI 部品に相当する。OperatableItem の実体は Transition（遷移）もしくは InputItem（入力項目）である。従って、Transition には操作可能な GUI 部品を、InputItem には入力可能な GUI 部品を割り当てる。本支援ツールでは Transition が ButtonComponent に、InputItem が InputComponent に対応している (図??)。

ButtonComponent は操作可能なボタンとして扱われる。また、内部的には Transition の他に UserIntention を持つ。これはどのような意図を持って画面遷移を行ったかを示すもので、シナリオの構成に必要な情報である。さらに、ButtonComponent は押されたとき、Browser へ押されたことを通知する。これにより、画面遷移を実現する。

InputComponent は複数の GUI 部品の集合として扱われる。なぜなら、内部に複数の InputItemInterface（入力項目）を持つためである。これは InputType および InputItemInstance に相当し、入力方法の指定、値の格納を担う。これにより、特定の画面内の一種類の項目に対して複数の値の格納が可能になる (図??)。また、入力項目を追加するボタン、指定入力項目を削除するボタンも InputComponent に格納される。

InputItemInterface は InputType に従い、以下の三種類の実装を持つ。

InputTextComponent（文字列入力） InputText に対応し、文字列入力可能な部品として扱われる。

MultipleSelectionComponent（複数選択） MultipleSelection に対応し、選択および文字列入力可能な部品として扱われる。

SingleSelectionComponent（単一選択） SingleSelection に対応し、選択および文字列入力可能な部品として扱われる。

現在、選択肢は画面遷移モデル上で定義されていない為、選択項目は文字列入力可能な部品になる。

GUI の構成は ComponentManager が担い、Screen（画面）から ButtonComponent および InputComponent を生成する。レイアウトは上段に入力項

目, 下段に操作項目を配置する (図??). また, 配置する際に InputComponent に元の InputItem.name をラベルとして付加する.

4.2 シナリオ作成に必要な要素

4.3 シナリオの導出

5 評価

5.1 評価方法

5.2 考察・結果

6 結論

6.1 まとめ

6.2 今後の課題