

XAML を利用した情報機器向け 2D/3DUI プロトタイプモデル生成システムの開発研究

北海道大学 ○樋口 大河, 金井 理

XAML-based 2D/3D UI prototyping system for the development of information appliances

Graduate School of Information Science and Technology, Hokkaido University

Taiga Higuchi, Satoshi Kanai

近年、情報機器において、ユーザビリティを考慮した UI ソフトウェアの効率的開発が求められている。その為には、設計初期に素早く評価できる 2DUI プロトタイプと、3D 筐体データ上でよりリアリティのあるシミュレーションが行える 3DUI プロトタイプとが統合的に設計できるシステムが要求されている。本研究では、状態遷移モデルから、UI の動的挙動を記述する XML 形式である XAML-B モデルを対話的に生成できるシステムを開発したので報告する。

1 はじめに

近年、デジタルカメラ、携帯電話などの情報機器においても UI (ユーザインタフェース) ソフトウェアの効率的開発方法が求められている。更に情報機器の多機能化により、開発段階でプロトタイプを使ったユーザビリティを考慮した設計も必要となってきた。これまでに情報機器 UI ソフトウェアの開発支援するツールとして DENIM [1], d.tools [2] 等が提案されている。しかしこれらを用いたプロトタイプシステムでは、UI 画面挙動の状態遷移モデル化に基づいたプロトタイプと UI シミュレーション機能を実現しているものの、UI 画面構造を陽に表現する静的構造モデルや、筐体デザインと 3D 形状デザインを統合した 3DUI シミュレーション機能、更にユーザビリティテスト・評価機能は、提供されていない。

そこで本研究では、これまでに筐体の 3D 形状モデルと、UI 画面構造やデザインの静的構造を統合的に記述できる XAML モデル [3] とを組み合わせ、更に画面遷移の動的挙動を記述することができる XAML-B モデルを独自に提案してきた。またこれらを用いて、情報機器の UI 操作シミュレーションが 2D, 3D いずれでも行える UI プロトタイプシステムを開発してきた [4] (図 1 参照)。しかしながら、UI の動的挙動を定義可能なシステムがないまま、XML 文書上に直接挙動の記述を行っていた。そこで、本報告では、動的挙動の XML 形式である XAML-B モデルデータを対話的に生成できるシステムを開発したので報告する。

2 開発したプロトタイプシステムの概要

前報では、XAML モデルと連携のとれた、UI の動的挙動を宣言的に記述できる XAML-B モデルを独自に提案し、その実行処理システムは既に開発済みである。しかし、これまで XAML-B モデルの動的挙動部分を定義するシステムが開発されていなかった為、ビジュアルに挙動を確認しながら挙動設計を行うことができず、定義の作業が繁雑でミスが発生しやすかった。

そこで本研究では、状態遷移モデルをビジュアルに設計確認しながら、イベントベースモデルに容易に変換でき、更に UI の動的挙動を記述した XAML-B モデルを XML 形式で保存できる UI 動的挙動定義システムを開発することとした。

3 UI 動的挙動定義システム

3.1 UI 動的挙動定義システムの概要

図 2 に提案する動的挙動定義システムの概要を示す。UI の動的挙動は、状態遷移モデルを用いて設計する初期動的挙動

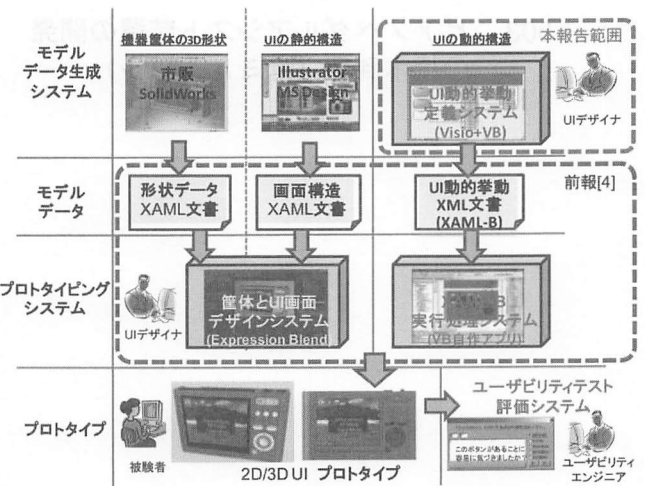


図 1: 2D/3D 統合型 UI プロトタイプシステム

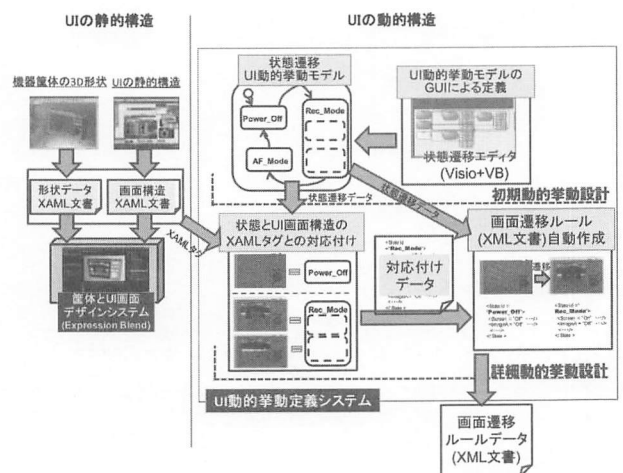


図 2: 動的挙動定義システム

設計と、イベントベースモデルによる詳細動的挙動設計の二つのプロセスで設計することとした。これは状態遷移モデルを用いた挙動記述は、UI の画面遷移の全体的な構造や手順をとらえやすく、開発上流段階での記述に適してからである。しかし、UI 開発が進み、動的挙動の詳細化が進むにつれて、状態の爆発が起こることが多い為、詳細挙動はイベントベースモデルで挙動設計することが望ましいからである。更に XAML モデル自身は陽に状態を持たず、UI の動的挙動はイベ

ント入力に対する各 UI 画面の上の部品の属性値変化の集合であるイベントベースモデルとして表現しなくてはならないからである。

動的挙動定義システムでは、図 2 のようにまず状態遷移エディタを用いて statechart 形式の状態遷移データを作成する。この状態遷移データと、UI 画面の静的構造のデザインが記述された XAML 文書とを組み合わせ、各状態と画面構造の XAML タグとの対応付けを行う。この対応付けされたデータの集合と状態遷移データとを用いて、画面遷移ルールを自動作成する。これらを XML 形式で保存することで XAML-B モデルデータを生成する。

3.2 初期動的挙動設計プロセス

初期動的挙動設計プロセスでは、UI デザイン初期の設計で用いられる画面遷移図と同等の表現を記述することができる状態遷移モデルを、statechart を用いて、挙動の設計をする。この作業は、本研究で開発した Visio と Visual Basic を組み合わせた状態遷移エディタを用いて行う。

3.3 詳細動的挙動設計プロセス

3.3.1 状態遷移モデルと XAML との対応付け

次に初期動的挙動設計プロセスで定義された状態遷移データと UI の静的構造で設計された XAML 文書とを用いて、状態遷移モデルと XAML タグとの対応付けを行う。図 3 に状態と UI 画面構造の XAML タグとの対応付けの例を示す。

まず図 3 の①のように、状態遷移エディタで定義された状態と、UI 画面構造の XAML タグとその属性値との対応付けを行う。状態遷移エディタで定義された状態名を持つ State タグを定義し、その下位のタグに、その状態が示す UI 画面のタグ集合を含める。例えば、Power_Off 状態を状態遷移エディタで定義した場合には、その Power_Off 状態に相当する UI 画面の XAML タグとその属性値を、Power_Off 状態のタグの下位内に含める。

次に図 3 の②のように、状態遷移エディタで状態が定義されなかった UI 画面と状態の対応付けを行う。まず設計した UI 画面構造が、どの状態の下位に含まれるかを定める。次にこの定められた状態名を持つ State タグを定義し、その下位のタグ内に UI 画面のタグ集合を含める。例えば、状態遷移エディタで定義されなかった Macro_Mode に対応した UI 画面が設計されているとすると、この Macro_Mode が上位にある状態と考えられる Rec_Mode という状態名を持つ State タグを新たに定め、その下位に Macro_Mode を規定する XAML タグとその属性値を含める。

これらの二つのプロセスにより、状態を陽に持つ場合と持たない場合両方において、UI 画面構造の XAML タグとその属性値の集合に、状態遷移モデルが関連づけられた対応付けデータを得られる。

3.3.2 画面遷移ルールの自動生成

XAML-B で UI 挙動の中心である画面遷移を表すイベントベースモデルは、XAML タグ内の属性値を変化させるルールの集合として記述できる。図 4 に示すように、このルールは上述の対応付けデータを用いて以下の手順で自動生成できる。初めに、図 4 の①に示すように、3.3.1 節で得られた対応付けデータを、同じ状態名を持つもの同士でまとめ分類する。例えば異なる UI 画面が同一の Rec_Mode という状態名の下位として定義されているならば、Rec_Mode 親状態を新たに生成し、複数の UI 画面はこれらの小状態として定義し直す。更にこれ

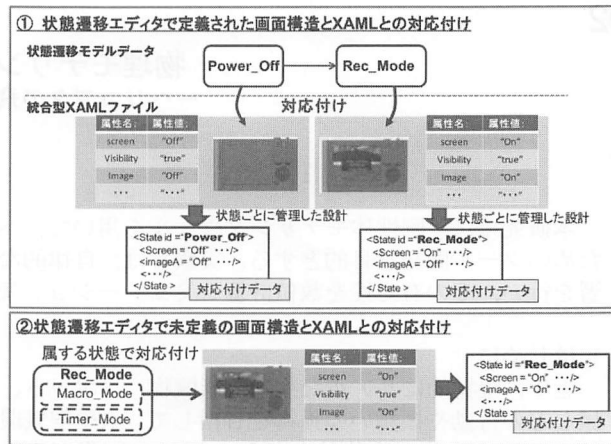


図 3：状態遷移モデルと XAML の対応付け

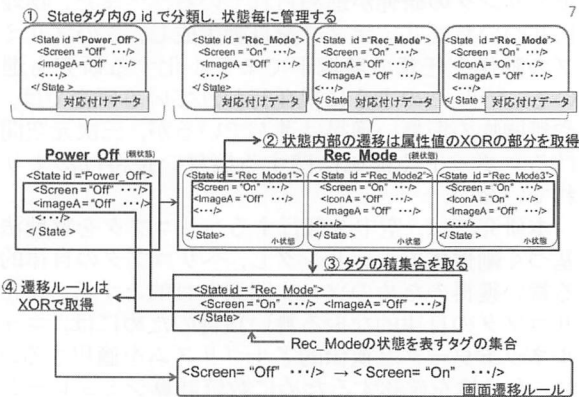


図 4：画面遷移ルール



図 5：3DUI シミュレーション

らの小状態の中でデフォルトの初期状態となるものをユーザが選択する。次に図 4 の②に示すように、これらの小状態間の画面遷移ルールをまず自動生成する。この画面遷移ルールは、小状態が持つ属性値の XOR で導出できる。次に、図 4 の③に示すように、親状態間の遷移を得る為に、最初に小状態に共通する属性値、つまり属性値の積集合を取る。この積集合は親状態(Rec_Mode の状態)を表すタグとその属性値となる。更に図 4 の④に示すように、既に初期挙動設計において状態遷移エディタで定義された遷移データを用いて、ある遷移に関する開始状態と終了状態を示す属性値の XOR を取得すれば、親同士の遷移を記述することができる。

得られた属性値の変化ルールを XAML-B 形式に整形し、これらを XML 文書で保存する。これらを既存の XAML-B 実行処理システムで読み込むことで、図 5 に示すように、動作可能な UI シミュレーションを行うことができた。

参考文献

- [1] M.W.Newman et al. "DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice." Human-Computer Interaction, 2003. 18(3): pp. 259-324.
- [2] B.Hartmann et al. "'Reflective physical prototyping through integrated design, test, and analysis", UIST2006, pp.299-308, 2006
- [3] 川西裕幸, 高橋忍 著:「XAML プログラミング」, ソフトバンククリエイティブ社, 2007.
- [4] 樋口大河, 金井理, XML を用いた携帯型情報機器向け 2D/3DUI プロトタイピングシステムの開発, 精密工学会春季大会学術講演論文集, B32, 2008.