画面遷移モデルに着目した ユーザビリティパターン抽出手法の提案

矢澤 幸也[1,a)]　小形 真平[1,b)]　岡野 浩三[1,c)]　海谷 治彦[2,d)]　鷲崎 弘宜[3,e)]

概要：ユーザビリティを効率的に高めるための Data Validation や Undo などの機能がユーザビリティパターンとして整理されている．しかし，そのパターンは今後の技術発展により変化しうるため，その時々にパターンを整理できるように，パターンの抽出手法を整備すべきであるが，未だ十分な手法はない．そこで，本稿では画面遷移モデルを対象としたユーザビリティパターンの抽出手法を提案する．そのキーアイディアは，既成の抽出対象モデルと，提案手法内で作成する抽出基準モデルとの差分からパターンを捉えることにある．抽出基準モデルはユーザビリティが考慮されにくいように，ユースケース毎に概念モデルに基づいて作成される CRUD 様から系統的に作成される．提案手法の有効性は，複数の事例モデルへの適用を通して評価する．

1. はじめに

近年，パソコンやスマートフォンなどの情報処理機器の急速な普及に伴い，システムのユーザビリティ [1] の重要性が高まっている．そのため，システムのユーザビリティを向上させるために，ユーザビリティを高める機能を導入する必要がある．例えば，業務システムにおいて効率性を高めるために，ユーザの操作ミスを防ぐための Data Validation や，実行した処理を元に戻すための Undo といった機能が導入される．このような機能はユーザビリティパターン [2] として整理され，それを活用して開発の一部を効率化する研究がある [3], [4].

ユーザビリティパターンの再利用によりシステムのユーザビリティを効率的に高められることが見込めるが，パターンは今後の技術発展により変化しうるため，その時々に整理できる手法を検討する必要がある．例えば，事例モデルベースのパターン整理手法の一つに，アナリシスパターンの自動抽出手法 [6] がある．当該手法では，対応する要求とモデルの構造および語の類似性に基づいてパターンを抽出・整理できるが，ユーザビリティに注目しておらず，ユーザビリティパターンの整理まではできない．

ユーザビリティパターンの抽出手法ではパターンの活用を見越して，(1) ユーザビリティパターンを獲得できる，(2) ユーザビリティパターンを形式化するために，機能に影響を与えるパラメータや機能構成を洗い出せる，(3) ユーザビリティパターンの再利用に向けて，パターンの導入手法を分析できる，ことが重要である．

本稿では，事例モデルからユーザビリティパターンを抽出する手法を提案する．事例モデルは，分析・設計仕様の一つである画面遷移モデル [5] とする．その理由は，画面はユーザビリティを直観的に判断しやすく，ユーザビリティによる影響が現れやすいと考えるためである．提案手法では，抽出対象モデルと抽出基準モデルとの差分からユーザビリティパターンを抽出する．抽出対象モデルとは，既成の分析・設計仕様のうち開発者が自由に作成した画面遷移モデルを指す．また，抽出基準モデルとは，ユーザビリティが考慮されないように，概念モデル [7] とそのユースケース毎の CRUD(Create，Read，Update，Delete) 仕様に基づき系統的に作成する画面遷移モデルを指す．

提案手法が前述の (1) － (3) を満たすかどうかを評価するために，複数の事例モデルに適用した．その結果，提案手法により既存のユーザビリティパターンに加え，整理されていないパターンを抽出できる見込みを得た．また，ユーザビリティパターンの活用に向けて，抽出したモデルからパターンに必要なパラメータや画面上での機能構成を洗い出せることを確認した．

[1] 信州大学
Shinshu University
[2] 神奈川大学
Kanagawa University
[3] 早稲田大学
Waseda University
[a)] 17w2094c@shinshu-u.ac.jp
[b)] ogata@cs.shinshu-u.ac.jp
[c)] okano@cs.shinshu-u.ac.jp
[d)] kaiya@kangawa-u.ac.jp
[e)] washizaki@waseda.jp

2. 用語説明

2.1 ユーザビリティパターン

ユーザビリティパターンとは，ユーザビリティを高める機能の構成をパターンとして整理したものであり，既存研究 [2], [3], [8] によりまとめられている．そこでは機能の概要や，システムに導入する目的などが記述され，それぞれの用途がわかるようになっている．以下に，整理されている計 18 種のユーザビリティパターンを示す．説明文上ほぼ同様の機能と解釈できたユーザビリティパターンについては，名称は異なるが同種のパターンとして併記する．

(1) Action for Multiple Objects[2]
2 つ以上のオブジェクトに対して一度に処理を実行できる．

(2) Auto Save[3]
ユーザが行った変更を自動的に保存する．

(3) Command Aggregation[2][8]，Scripting[3]
一度に複数のタスクを実行するためのコマンドを作成できる．

(4) Context Sensitive Help[2]
ユーザの行っていることを監視し，そのタスクに関係するヘルプを提供する．

(5) Data Validation[2]，Live Validation[3]，User input error prevention / correction[8]
フォームやフィールド内の入力データが正しく入力されているか検証する．

(6) Emulation[2]
ユーザの使い慣れたインタフェースや動作を模倣する．

(7) History Logging[2]
ユーザの操作ログを記録する．

(8) Multiple views[2]
用途に合わせて選択できる複数のビューを提供する．

(9) Preview[3]
アクションを実行した際に期待される結果のプレビューを提供する．

(10) Recycle Bin[3]
削除を選択したデータを完全に削除せず，要求に応じて復元できるように仮想のゴミ箱に移動する．

(11) System Feedback[2]，Feedback[8]，Progress Display[3]
変更のあったシステムのデータや，システムの処理の進捗といった状態をユーザに伝える．

(12) Undo / Cancel[2][3][8]
ユーザが実行した操作または進行中の操作をキャンセルできる．

(13) User Modes[2]
ユーザの能力に応じて機能が調整できるように，例えば，ビギナー用，ミドル用，エキスパート用といった複数のモードを提供する．

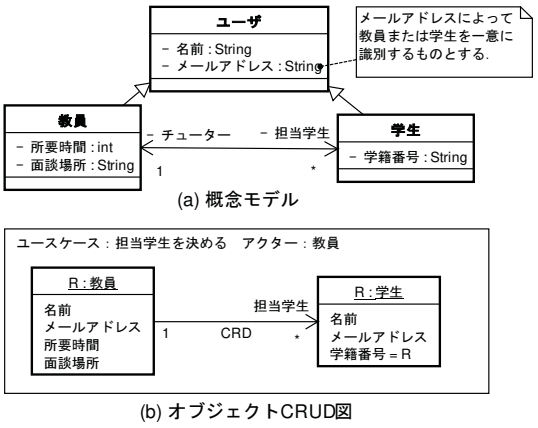

図 1　概念モデルとオブジェクト CRUD 図
Fig. 1　A conceptual model and an object CRUD diagram.

(14) User Profile[2][8]
ユーザインタフェースのレイアウトや表示するデータ量などをユーザ毎に設定できるように各ユーザのプロファイルを作成および記録する．

(15) Wizard[2][8]
複数のステップを含むタスクを一連のサブタスクに分割する．

(16) Multi-Channeling[2]
デスクトップによる入力や，音声入力，TV からの入力といった異なるタイプの入出力デバイスを使用したアクセスを許可する．

(17) Reuse information[8]
ファイルを別のフォルダにドラッグ&ドロップで移動できるように，ユーザがシステムのデータをある場所から別の場所に移動できる．

(18) Workflow model[2]
複数の異なるユーザがワークフローに従って一つのデータを適切に作成できるように，適切な順序で必要な部分データの操作を行える機能を提供する．

本研究では，上記のパターンを参考に，抽出対象モデルからユーザビリティパターンを抽出する．ただし，画面遷移モデルに記述される画面構造または画面遷移に影響が及ぶと考えられる（1）－（15）を提案手法で扱うものとする．

2.2 概念モデル

概念モデル [7] とは，システムやデータの概念構造を整理するモデルである．図 1(a) は概念モデルの例であり，大学の教員・学生向けに作成されたチューター面談予約システムのモデルである．本システムは，各教員が複数学生のチューターとなり，定期的に実施される面談のスケジュールを管理するためのものである．なお，紙面の都合により図 1 は，教員がシステムに登録された学生から担当学生を選択するユースケース（担当学生を決める）に限って抜粋したものである．図 1(a) では，教員が担当学生を選択する

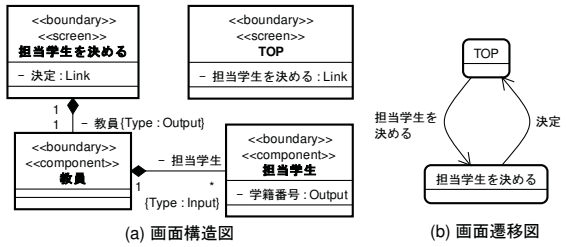

図 2　画面遷移モデル
Fig. 2　A screen transition model.

ために学籍番号を参照するため，学生クラスに “学籍番号” 属性が記述されている．本研究では，提案手法において概念モデルを応用して抽出基準モデルを作成する．

2.3　オブジェクト CRUD 図

オブジェクト CRUD 図 [9] とは，ユースケース・アクタ単位で概念モデルを基にオブジェクトの CRUD(Create，Read，Update，Delete) を表す図であり，オブジェクト図の記法をベースに記述する．オブジェクト図において，インスタンス仕様の分類子（Classifier）名，スロット名およびリンク名は概念モデルを基に名前を決定し，インスタンス仕様名とスロット値，リンクは CRUD を記述する．

オブジェクト CRUD 図の例を図 1（b）に示す．教員による学籍番号を介した担当学生の選択に基づいて，システムは教員と学生のインスタンスを読み込み，その間にリンクを作成してチューターと担当学生を結びつける．そのため，オブジェクト CRUD 図には教員と学生のインスタンス仕様が記述され，各インスタンス仕様と学籍番号に R が，リンクに C が記述されている．リンクに C の他に RD があるのは，教員が担当学生を確認することや，担当学生の変更によりリンクを削除するためである．本研究では，抽出基準モデルを作成するベースとしてオブジェクト CRUD 図を使用する．

2.4　画面遷移モデル

ユーザビリティパターンの多くはユーザインタフェース上のデータ構造や機能構成に表れると考えられる．したがって，ユーザインタフェースの分析・設計に用いられる画面遷移モデルを抽出対象モデルおよび抽出基準モデルとして扱う．

画面遷移モデル [5] とは，画面の入出力項目や画面の遷移を表したモデルであり，クラス図による画面構造定義とステートマシン図による画面遷移定義がある．本稿では，それぞれを画面構造図と画面遷移図と呼ぶ．図 2 に画面遷移モデルの例を示す．

2.4.1　画面構造図

画面構造図はクラス図により画面の入出力項目や遷移といった画面の構造を定義する．クラスには，ユーザとシステムの境界を示す boundary をステレオタイプに記述する．加えて，screen か component のどちらかをステレオタイプとして与える．screen は画面を表し，component は画面の一部を表す．図 2(a) の場合，教員が担当学生を決める際に複数の学籍番号を入力するため，コンポジションの関係で表されている．

属性の型には Input，Output，Link のいずれかを指定する．Input は画面遷移を生じさせない入力項目，Output は文字列などの出力項目，Link は画面遷移を生じさせる入力項目であることを表す．また，component のコンポジションに Type として Input あるいは Output を指定することで，該当の component をオブジェクト単位で入出力することを表す．

2.4.2　画面遷移図

画面遷移図はステートマシン図の状態を画面とみなして画面間の遷移を定義する．状態は画面構造図の screen と一対一で対応させ，状態間の遷移は screen 間の遷移として表現する．また，トリガ名は screen に定義された Link 型の属性名と対応付ける．なお，遷移の矢印の方向は screen の遷移先を表す．図 2(b) の場合，図 2(a) の “TOP” と “担当学生を決める” の 2 つの screen が状態で記述され，それぞれの Link 属性が遷移としてトリガ名 “決定” と “担当学生を決める” で記述されている．

3.　提案手法

3.1　アイディア

抽出対象モデルと抽出基準モデルとの差分からユーザビリティパターンを抽出する．そこで，抽出基準モデルにユーザビリティが考慮されないように，概念モデルで表す構造はユースケースの達成に必須であることから，そこにはユーザビリティの考慮が及び難いと仮定する．また，オブジェクト CRUD 図は概念モデルに沿った振舞いを表し，開発者がユーザビリティを考慮した仕様になりにくいと考えられる．そこで，オブジェクト CRUD 図から画面遷移モデルへの自動生成手法 [9] を拡張して抽出基準モデルを系統的に作成する．

次に，モデルの差分から得られたユーザビリティパターンのモデル（以降，抽出モデルと呼ぶ）を 2.1 節のユーザビリティパターンと対応付け，既に整理されているパターンか，未だ整理されていない新規のパターンかを見極める．また，ユーザビリティパターン毎に抽出モデルの特徴を整理し，パターンの機能構成や導入手法を調査する．ただし，既存のユーザビリティパターンと抽出モデルを対応付ける際に，ユーザビリティの知識が必要となるため，提案手法の利用者はユーザビリティの専門家とする．

3.2　概要

抽出対象モデルからユーザビリティパターンを抽出する

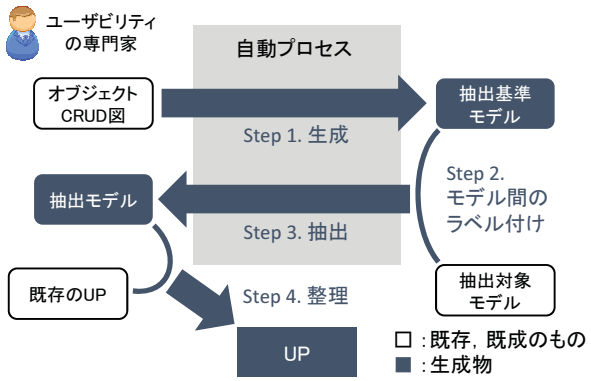

図 3　提案手法の概要

Fig. 3　The overview of the proposal method

手法の概要を図 3 に示す．業務システムにおいて，ユーザの操作や管理状況を目視で確認できる画面は重要であり，従来からその画面遷移や画面構造はモデルを使用して設計されている [10], [11]．また，ユースケースとデータの関係を定義するために CRUD 分析も従来から行われている [12] ことから，それらが開発時によく行われると考えられる．したがって，提案手法では開発の分析・設計時にオブジェクト CRUD 図および画面遷移モデルを記述するプロセスが含まれていることを前提とする．

ユーザビリティパターンの抽出手法のプロセスは図 3 の 4 つの Step で構成される．自動プロセスの枠にある 2 つのステップは系統的な生成プロセスであり，残りの 2 つは手動プロセスである．以下に各ステップの概要を示す．

Step 1.　抽出基準モデルをオブジェクト CRUD 図から生成する．

Step 2.　抽出対象モデルと抽出基準モデルの間で対応付くモデル要素を明確にするために，対応関係が分かるようにモデル要素にラベル付けを行う．

Step 3.　抽出基準モデルと対応付かない抽出対象モデルのモデル要素を抽出モデルとして抽出する．

Step 4.　既存のユーザビリティパターンの特徴から抽出モデルを整理して，ユーザビリティパターンのパラメータを抽出する．

3.3　Step 1. 抽出基準モデルの生成

オブジェクト CRUD 図から抽出基準モデルとする画面遷移モデルを生成する．抽出基準モデルは，オブジェクト CRUD 図から画面構造図を生成し，その後に画面構造図を基に画面遷移図を生成する．図 4 にオブジェクト CRUD 図から抽出基準モデルの生成例を示す．

3.3.1　画面構造図の生成

画面構造図は，表 1 と表 2 を基に生成する．表 1 のスロットに関する対応関係以外は生成手法 [9] を拡張した対応関係であり，それに従って画面構造図の screen と component を生成する．例えば，図 4 では，オブジェクト CRUD 図の

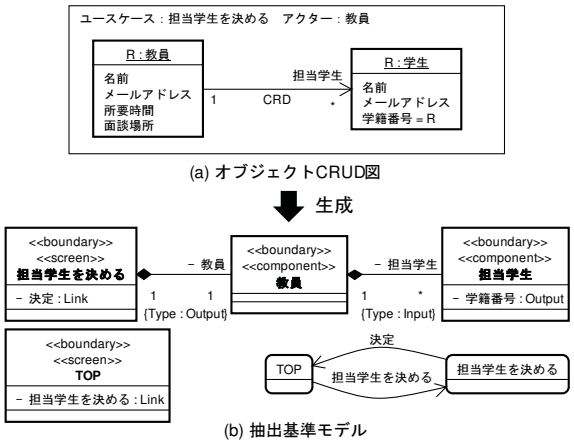

図 4　オブジェクト CRUD 図から画面遷移モデルの生成

Fig. 4　Generating a screen transition model from object CRUD diagrams.

表 1　オブジェクト CRUD 図と画面構造図の対応関係

Table 1　Correspondence between object CRUD diagrams and a screen structure diagram.

オブジェクト CRUD 図	画面構造図
ユースケース	screen
ユースケース名	screen のクラス名
インスタンス仕様	component
リンク	コンポジション
リンク名 (C，U，D)	Type（Input）
リンク名 (R)	Type（Output）
リンク端名	component のクラス名
リンク端名	コンポジションの関連端名
リンクの多重度	コンポジションの多重度
スロット名	属性名
スロット値（C，U，D）	属性の型（Input）
スロット値（R）	属性の型（Output）

ユースケース名を基に抽出基準モデルの “担当学生を決める”screen が生成され，教員と学生インスタンスは “教員” と “担当学生”component に変換されている．ただし，“教員”cmponent のように，screen とコンポジションの関係にある component のクラス名はインスタンス仕様の分類子名とする．

次に，screen と component 間，component と component 間のコンポジションを引き，それに付随するモデル要素を追加する．screen とコンポジションの関係になる component は，リンクの元を辿って最終的に辿り着いたインスタンス仕様とし，コンポジションの関連端名はインスタンス名に，Type はインスタンス仕様の分類子名（C，U，D であれば Input，R であれば Output）に従う．該当するインスタンス仕様が複数ある場合は，screen からそのすべての component にコンポジションを引く．また，リンクがループしている場合は，インスタンス仕様のうち書き込み（CUD）の多さ，次いで読み込み（R）の多いものに screen へのコンポジションを引く．他のコンポジションとそれに

表 2　定型的な作成要素
Table 2　Fixed elements embedded in a screen structure diagram.

図要素名	説明
TOP 画面	クラス名が TOP の screen を作成する．
TOP 画面から各ユースケースへのリンク	TOP 画面から各 screen への遷移を表す Link 属性を作成する．属性名は各 screen のクラス名とする．
各ユースケースから TOP 画面へのリンク	ユースケースを達成した際に TOP 画面へ戻るための遷移を表す Link 属性を，TOP 画面を除いた各 screen に追加する．追加する属性は “決定：Link” とする．

付随するモデル要素は，表 1 に従う．ただし，リンク名について書き込みと読み込みが併用されている場合は Input とする．図 4 では，教員と学生インスタンスのうち教員インスタンスがリンク元であるため，screen と教員インスタンスの component 間にコンポジション引かれている．また，“担当学生”component の Type は，リンクが CRD であるため，Input が指定されている．

そして，画面構造図の各クラスに属性を追加する．例えば，図 4 では，オブジェクト CRUD 図の “学籍番号 = C” が画面遷移モデルの担当学生クラスの “学籍番号：Input” に変換されている．ただし，スロット値について書き込みと読み込みが併用されている場合は属性の型を Input とする．

最後に，表 2 の定型的な生成要素を追加する．図 4 の抽出基準モデルでは，TOP 画面の screen に “担当学生を決める：Link” が作成されている．また，“担当学生を決める”screen に “決定：Link” が作成されている．

3.3.2　画面遷移図の生成

画面遷移図の作成手法は先行研究 [9] の手法に従う．まず，ステートマシン図において画面構造図の screen 毎に screen 名（例：担当学生を決める）の状態を記述し，TOP 画面の状態から TOP 画面を除いた各 screen の状態に一対一で対応するように遷移を引く．トリガーには該当の Link 属性名を記述する．例えば，図 4 では TOP 画面から担当学生を決める画面に対して，トリガ名 “担当学生を決める” という遷移を引いている．次に，TOP 画面を除く screen ユースケースを達成した際に TOP 画面に遷移するための遷移を引く．トリガー名は “決定” とする．以上の操作で画面遷移図を生成する．

3.4　Step 2. モデル間のラベル付け

抽出対象モデルは画面名や属性名といった名称を制限なしに自由に定めている可能性が高いことから，ユーザビリティパターンの抽出を行う前に，抽出対象モデルと抽出基準モデル間で一致するモデル要素を識別する．モデル要素の識別する工程の例を図 5 に示す．

抽出対象モデルのうち抽出基準と対応するモデル要素に目印として任意のラベルを付ける．例えば，図 5 では，抽出対象モデルの “面談可能日時を選択する” と抽出基準モデルの “面談可能日時の選択” が対応するため，ラベルとして “S1” を付けている．同様の操作により，対応する screen，component，Type，属性，遷移のリンク名にラベル付けを行う．ただし，ラベルの混同を回避するためにモデル要素間の対応関係毎に異なるラベルを付ける．

3.5　Step 3. 抽出モデルの抽出

Step 2 によりラベルが付加されていないモデル要素は，抽出基準モデルと抽出対象モデルとの差分であり，それを抽出モデルとして抽出する．ただし，各抽出モデルがどのユーザビリティパターンであるかはわからないため，ここでは抽出モデルをユーザビリティパターンの候補とする．

抽出は，複数の属性やクラスからなるユーザビリティパターンの分解を回避するために，（1）screen 単位，（2）component 単位，（3）属性単位，の順に行う．図 6 はその抽出例である．

ラベルが付加されていない screen クラスは，一つずつ抽出する．ただし，ラベルが付加されていない screen 間に遷移があり，それらの screen 間にわたって抽出基準モデルの複数の属性が存在する場合は，該当する screen をまとめて抽出する．また，遷移にラベルが付与されていない場合は，その遷移に関係する screen と共に抽出する．例えば，図 6(a) において，ラベルが付与されていない “空き日時”，“予約の催促”，“面談情報の設定” の遷移を，図 6(b) のように関係する screen と共に抽出している．

component 単位では，ラベルが付加されていない component を一つずつ抽出する．例えば，図 6(c) では，リンクメニューの component にラベルが付加されていないので，component がそのまま抽出されている．ただし，component に対して component が集約されている場合は，それらをまとめて抽出する．また，component にラベルが付加されている場合でも，Type にラベルが付加されていなければ，そのコンポジションに関係する screen または component をまとめて抽出する．

ラベルが付加されていない属性は，その属性単体を抽出する．図 6(c) では，“担当学生入力手法” 属性にラベルが付加されていないため，図 6(e) のように抽出している．

3.6　Step 4. 抽出モデルの整理

抽出モデルがどのユーザビリティパターンであるかを識別するために，2.1 節のパターンの特徴と抽出モデルを照合する．例えば，Wizard は “複数のステップを含むタスクを一連のタスクに分割する” と記述があることから，抽出

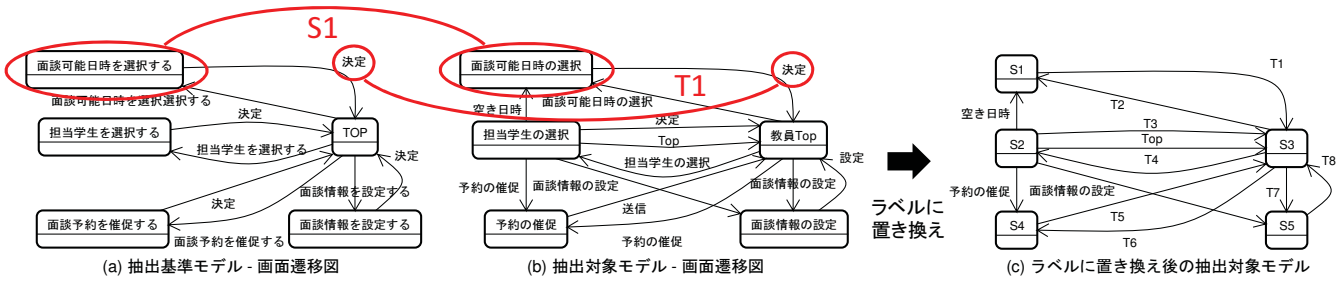

図 5　抽出対象モデルへのラベル付与

Fig. 5　Labeling an extraction target model.

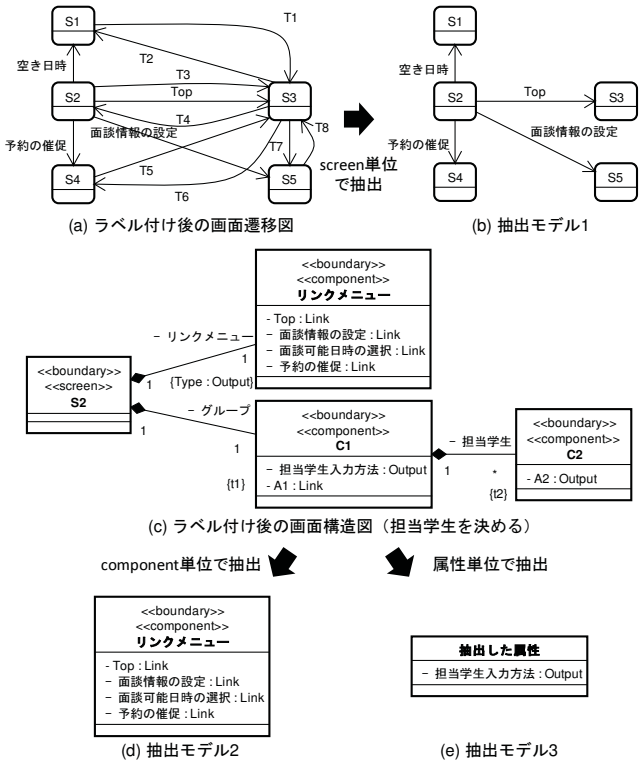

図 6　抽出モデルの抽出

Fig. 6　Extracting the extracted models.

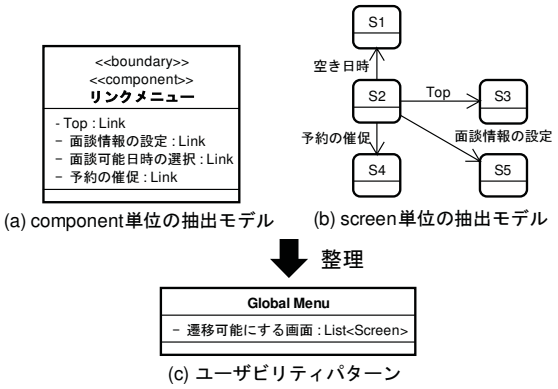

図 7　抽出モデルからのユーザビリティパターンの整理

Fig. 7　Arranging usability patterns from the extracted models.

基準モデルにおいて一つの screen に複数ある Input 属性が，抽出対象モデルにおいて複数の screen にわたって存在する場合，その抽出モデルを Wizard とみなす．

次に，ユーザビリティパターン毎に抽出モデルの共通の属性やモデル構成を整理し，機能に影響を与えるパラメータを，そのパターンの属性としてモデルで定義する．図 7 は，モデルの定義例である．図 7 の (a) と (b) は共通のユーザビリティパターンを表し，screen 間の遷移に関係した機能である．任意の screen の遷移を可能にするために，遷移可能にする画面を選択する必要があるため，“遷移可能にする画面：List<Screen>” 属性を定義している．

既存のユーザビリティパターンと一致しない抽出モデルのうち，図 6(e) のような属性一つで構成され，かつ型が Output のものは説明文や操作案内といったものが多く，機能とは関係し難いと考えられる．そのため，前述のような属性は抽出モデルから除く．残った抽出モデルは新たなユーザビリティパターンとして定義する．

3.7　手法の限界

提案手法では，ユーザビリティが考慮されにくいモデルとして，概念モデルと，その振舞いをユースケース毎に表現するオブジェクト CRUD 図を採用しているが，それらの仕様が業務の内容を忠実に再現している場合でも，ユーザビリティ側面が含まれる可能性がある．そのため，抽出基準モデルがユーザビリティを考慮していないものとして保証できない．また，抽出基準モデル生成時の定型的な作成要素に関しても，一定のユーザビリティが考慮されている可能性がある．

エンドユーザは与えられた画面の操作により，個別に異なるユーザビリティを感じることになる．そのため，そもそもユーザビリティを全く考慮していないと保証することは困難を極める．現実的には，開発プロジェクト毎に，手持ちの GUI ツールキットやフレームワークなどに応じて，ベースラインとなる画面構成を決められる柔軟性が重要であると考えるが，この柔軟性の考慮は今後の課題となる．

4.　ケーススタディ

4.1　概要

事例モデルに提案手法を適用し，以下の 3 つを確認する．

表 3　抽出されたユーザビリティパターンの数
Table 3　The number of the extracted usability patterns.

	a	b	c	d	e	f	g	h	合計
Wizard	2	0	0	0	0	0	1	0	3
History Logging	2	0	0	0	0	0	0	0	2
Cancel	3	0	1	0	0	0	3	0	7
System Feedback	5	0	0	0	0	1	0	0	6
Global Menu	7	0	0	0	0	0	0	0	7
合計	19	0	1	0	0	1	4	0	25

(1) ユーザビリティパターンを獲得できるか.
(2) 機能に影響を与えるユーザビリティパターンのパラメータや画面上の機能構成を洗い出せるか.
(3) ユーザビリティパターンの再利用や系統的な導入に向け，パターンの導入手法を探れるか.

適用する事例モデルは（a）チューター面談予約システム，（b）チャットシステム，（c）スポット検索システム，（d）ネット掲示板，（e）リバーシ，（f）ATM システム，（g）IT 単語クイズ，（h）アルバイト代計算システム，の計 8 つのフルスクラッチで作成されたシステムモデルである．（a）と（b）は，情報工学系の大学 4 年次相当の学生が 3 名 1 グループにより作成され，画面遷移モデルの作成にあたってソフトウェア工学専門家からユーザビリティに関する内容を含めて 2，3 回程度のレビューを受けている．また，（c）－（h）は，情報工学系の短大 2 年次相当の学生が 3 名 1 グループが作成したモデルである．なお，画面遷移モデルの作成にあたってはソフトウェア工学専門家から 1 回のレビューを受けている．ただし，（c）－（h）は CRUD 分析がなされていないため，実験者がユースケース記述を解釈してオブジェクト CRUD 図を作成した.

4.2　結果

事例モデルへの手法適用によるユーザビリティパターンの抽出結果を表 3 に示す．表 3 における各列のアルファベットはそれぞれ 4.1 節の事例モデルに対応し，行は獲得したユーザビリティパターンを表す．行にあるユーザビリティパターンのうち，Global Menu は 2.1 節のパターンと対応付かなかったため，“各画面から任意のユースケース画面へ遷移できるリンクを提供する機能” として新たに定義した．セルの値は，各ユーザビリティパターンと対応付いた抽出モデルの数を表す．例えば，a の Wizard に 2 とあるが，これは事例モデル a の 2 箇所から Wizard の抽出モデルを獲得したことを意味する.

4.3　考察

4.3.1　ユーザビリティパターンの獲得

事例モデルへの手法適用の結果，複数のユーザビリティパターンが抽出され，そのうち既存のパターン集と一致しないパターン（Global Menu）が得られた．そのパターンは，Web サイトなどのメニューバーでしばしば使用されており，操作性の面でユーザビリティの向上に貢献することが考えられるため，ユーザビリティを高める機能であることが期待できる．このことから，提案手法により既存のユーザビリティパターンに加え，整理されていないパターンの獲得が見込める.

一方で，2.1 節のユーザビリティパターンのうち抽出されたパターンは 4 種類であり，抽出可能なパターン数の半分を下回る結果だった．これは，事例モデル作成者のほとんどが開発経験の浅い学生であったために，ユーザビリティがほとんど考慮されなかったことが原因と考えられる．したがって，より多くのユーザビリティパターンを獲得するためには，対象の事例モデルを選定する必要があり，より実用的なシステムモデルへの手法適用が今後の課題となる.

4.3.2　パラメータと機能構成の調査

ユーザビリティパターン毎に抽出モデルを整理することで，それらに共通点があることがわかった．例えば，表 3 からわかるように，Global Menu は 7 つの抽出モデルが得られており，それらの共通点として図 7(a) のように複数の Link が screen に含まれていた．その共通点は，ユーザビリティパターンの必須なパラメータであり，抽出モデルの構成は，ユーザビリティパターンの導入により構成された画面の部品であり，パターンの機能構成とみなせる．したがって，提案手法によりユーザビリティパターンの属性や機能構成を調査でき，その結果の整理によりパターンの形式化に繋げられることが期待できる.

4.3.3　ユーザビリティパターン導入手法の分析

抽出モデルが適用されていた箇所からユーザビリティパターンの導入箇所が明確になった．また，各抽出モデルの形態はユーザビリティパターンの画面遷移モデルへの適用例であり，それらの共通点を洗い出すことで，パターンの基本構成や派生したモデル要素を特定できた．その結果を基にユーザビリティパターンの導入手法を形式化することにより，画面遷移モデルへの系統的なパターン導入手法を検討できることが見込める．しかし，表 3 のように，History Logging の抽出モデルは 2 つしか抽出されていないため，基本構成の特定すら困難であった．ユーザビリティパターンの導入手法を形式化するには，モデルの基本構成や派生したモデル要素の一般化は必須であり，一般化できるだけの事例モデルの収集と手法適用が今後の課題となる.

4.3.4　手法の改善

Step 4 は手動プロセスであり，表 3 の事例モデル a のように多くの抽出モデルが得られるケースがあったため，手動では手間を要した．そこで，手法の効率性を高めるために，Step 4 を自動化する必要がある．その手法の一つとして，ユーザビリティパターン毎にその特徴からテンプレートのモデルを作成し，それと抽出モデルとのパターンマッチング [13] により自動的に分類する手法が考えられる.

5. 関連研究

野本ら [6] は，要求から優れたモデルを導出する過程のパターンを自動抽出する手法を提案しており，その手法により人手による抽出コストを削減できることが見込める．野本らの手法はパターンの抽出手法であり，本稿の提案手法と類似するが，ユーザビリティに着目していない．一方で，本稿の提案手法はユーザビリティに着目したパターンの抽出手法を提案している．そのため，提案手法により抽出したパターンを活用することでユーザビリティを高められることが期待できる．

Folmer ら [2] の研究は，ユーザビリティとソフトウェアアーキテクチャとの関係を把握するためのフレームワークを作成する研究である．その中で，従来のユーザビリティパターンを整理し，ユーザビリティとの関係を示している．ユーザビリティパターンとユーザビリティとの関係とは，パターンの導入によりどのユーザビリティ特性が向上あるいは低下するかを整理したものである．また，Roder[3] は，開発経験の浅い開発者でもユーザビリティパターンを適切にユースケース記述に組み込めるように，そのガイドラインとなるパターンの仕様テンプレートを作成している．本研究において抽出したユーザビリティパターンは，ユーザビリティ特性の向上・低下やその実例，仕様テンプレートの記述はないが，パターンの導入に必要な属性や機能構成を整理し，実用性を考慮したパターン定義がされている．したがって，パターンの導入に向けた指針として活用できる見込みがある．

Juristo ら [8] の研究は，ユーザビリティの専門家に依存することなく，開発者がユーザビリティ要求を獲得できるように，ユーザビリティパターン毎に仕様とそのパターンの要求抽出ガイドを作成している．Juristo らは，その作成のためにユーザビリティパターンを文献から収集しているが，本稿の提案手法では，事例モデルからパターンを抽出している．そのため，事例モデルからユーザビリティパターンの情報を収集でき，詳細にパターンを整理できる．

6. おわりに

本稿では，今後の技術発展によりユーザビリティパターンが変化した場合でも，事例モデルからパターンを整理できるように，画面遷移モデルに着目したユーザビリティパターン抽出手法を提案した．事例モデルに提案手法を適用した結果，提案手法により既存のユーザビリティパターンに加え，整理されていないパターンを獲得できる見込みを得た．また，ユーザビリティパターンに必要な属性や機能構成，適用箇所を抽出モデルから把握でき，パターン導入手法の調査やパターンの形式化に有用であることが示唆された．それらの結果を基に画面遷移モデルへの系統的なパターン導入手法の確立が期待できる．

今後は，抽出したユーザビリティパターンを活用して効率的な開発手法を確立することが展望となる．具体的には，抽出モデルの特徴をパターン別に整理してパターンを形式化することや，より実用的なシステムモデルへの手法適用により，ユーザビリティパターンを拡充することが課題となる．また，抽出したユーザビリティパターンのモデルを抽出基準モデルに適用し，元の抽出対象モデルを作成できるかどうかを実証する必要がある．

謝辞 本研究は，JSPS 科研費 JP16H02804 および JP15K15972 の助成を受けたものです．

参考文献

[1] ISO/IEC: *Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models*, ISO/IEC Std. 25010:2011 (2011).
[2] Folmer, E., Gurp, J. V. and Bosch, J.: *A framework for capturing the relationship between usability and software*, Software Process: Improvementand Practice (2003).
[3] Roder, H.: *Specifying usability features with patterns and templates*, UsARE (2012).
[4] 紙森翔平，小形真平，海尻賢二：モデル駆動開発におけるユーザビリティ機能を実装した Web プロトタイプの自動生成，SES (2014).
[5] Kamimori, S., Ogata, S. and Kaijiri, K.: *Automatic Method of Generating a Web Prototype Employing Live Interactive Widget to Validate Functional Usability Requirements*, ACIT-CSI (2015).
[6] 野本悠太郎，久保淳人，鷲崎弘宜，深澤良彰：構造および語の類似性に基づくアナリシスパターンの自動抽出，FOSE (2009).
[7] 児玉公信：情報システム設計における概念モデリング，人工知能学会誌 (2010).
[8] Juristo, N., Moreno, A. M. and Sanchez-Segura, M.-I.: *Guidelines for eliciting usability functionalities*, IEEE Trans. Softw. Eng. (2007).
[9] 小形真平，上森翔平，海谷治彦，岡野浩三：画面遷移モデリングにおける関心事の分離法の検討〜業務機能と使用性向上機能に着目して〜，KBSE (2015).
[10] 近藤恵一，森田武史，和泉憲明，橋田浩一，山口高平：エンタープライズアプリケーションオントロジーに基づく業務アプリケーション開発支援，人工知能学会論文誌 (2008).
[11] Ceri, S., Fraternali, P. and Bongio, A.: *Web Modeling Language (WebML): a modeling language for designing Web sites*, Computer Networks (2000).
[12] Brandon, Jr, Daniel: Crud matrices for detailed object oriented design, Journal of Computing Sciences in Colleges (2002).
[13] Ballis, D., Baruzzo, A. and Comini, M.: *A rule-based method to match Software Patterns against UML Models*, Electronic Notes in Theoretical Computer Science (2008).