シナリオ順応型 UI 設計ツール

草野 孔希　　中谷 桃子　　大野 健彦*

概要. 本研究ではシナリオを用いたユーザインタフェース（UI）設計を支援するツールを提案する．ユーザにとって使いやすい UI を実現するには「誰が，どのような目的で，どのように使うか」といった，ユーザの振る舞いを熟慮することが重要である．そこで，振る舞いを小説のようにシナリオとして記述することで，特別な知識がなくてもユーザ像を具体的にイメージすることが可能となる．しかし，多数の利用シナリオが考えられるインタラクティブシステムにおいて，シナリオ同士の関係性やトレードオフを加味しながら UI を設計することは難しい．そこで，本研究ではシナリオの階層化およびタグ付けを用いたシナリオの分析支援，および分析結果の可視化によって UI 設計を支援するツールを提案する．

1　はじめに

インタラクティブシステムにおいて，使いやすいユーザインタフェース（UI）を実現するためには，システムを「誰が，どのような目的で，どのように使うか」といった，ユーザがシステムを用いて目的を達成するまでの振る舞いや環境を熟慮して UI を設計することが重要である [4].

システムを用いてユーザがどのように振る舞うかを明確にするためには，図 1 に示したようなシナリオを作成することが有用である [2]．自然言語を用いたシナリオは豊かな表現が可能であり，専門知識がなくても理解が容易である．そのため，UI 設計者に限らず，異なる立場の関係者（プログラマ，顧客など）とも合意形成がしやすくなる．

しかし，ユーザ像を明確化し，シナリオによってユーザの振る舞いを具体化できたとしても，使いやすい UI を設計することは容易ではない．特に，UI 設計において，ユーザ像やシナリオから情報を正しく読み取り，UI に活かすことが難しいと指摘されている [15]．その理由の一つは，インタラクティブシステムでは複数の利用シナリオが考えられることにある．複数のシナリオがある場合，1 つのシナリオに最適な UI を設計できたとしても，他のシナリオでは使いにくい UI となってしまう．そのため設計者は，どのシナリオを優先するか，他シナリオとの共通点やトレードオフなどを分析する必要がある．さらに，それらの分析結果を加味した上で，どのような UI コンポネントを割り当て，配置するか，画面遷移はどうするかなどを検討する必要がある．シナリオを用いた UI 設計において，これらのシナリオから UI を検討する工程はいまだに十分な支援がなされておらず，UI 設計者のノウハウにゆだねられている困難で手間のかかる作業である．

ケータイで暇つぶしに動画を探して閲覧したい
電車の中で暇つぶしをしたいと思って，スマートフォンを取り出した．スマートフォンから動画アプリを起動すると，今日の**おすすめ動画が表示**されていた．**動画にはタイトルや再生数，サムネイルなどの情報が表示**されています．その中から，見たい動画を…

図 1. シナリオ例

また，使いやすい UI を設計するには，ユーザビリティ評価と UI 設計を反復することが重要である [7]．その際，設計時に予め作成したシナリオを，テストシナリオとしてユーザビリティ評価にも活用できる．シナリオによって，単純な使い勝手だけでなく，想定ユーザと実ユーザの振る舞いの差異を確認しながら，UI 改善の検討ができる．しかしその際に，修正対象の UI がシナリオのどこと関係するかを，設計者自身が確認しなければならない．加えて，修正対象の UI が複数のシナリオに影響を受けている場合もあるため，確認する作業は極めて面倒である．さらに，UI の評価と改善を反復するごとに確認作業が必要になるため，結果として UI とシナリオとの関係性の維持が次第に困難になり，最終的にはシナリオと UI とが乖離してしまう．

そこで本研究では，シナリオの分析およびその分析結果を用いた UI 設計の難しさを軽減するとともに，シナリオと UI との対応関係の維持を容易にする．これにより，システムエンジニア (SE) など，UI 設計の専門家でなくても，ユーザの要求を的確に捉えた，使いやすい UI を設計しやすくする．特に本研究では，PC やスマートフォンで動作するインタラクティブシステムの GUI に着目し，GUI 設計を二側面から支援する実用的なツールを実現する．第一にシナリオの分析を支援するために，シナリオの階層化をすることで要求を明確化し，それらの要求を満たすための機能を抽出しやすくする．第二に，シナリオの分析結果を GUI 設計に活かしやすくするために，分析結果を自動的に可視化する．これらの支援によって，設計者は，的確かつ素早くシナリ

* Koki KUSANO, Momoko NAKATANI and Takehiko OHNO, NTT サービスエボリューション研究所 ヒューマンアナリシスプロジェクト

オを用いた GUI 設計の反復が可能となり，使いやすい GUI を実現しやすくなる．

以降，2 章では本研究に関連する研究や技術について述べる．3 章では，提案するツールとその特徴について紹介し，4 章で実装について述べる．さらに，提案ツールについて 5 章にて考察し，最後に 6 章でまとめと今後について述べ，本論を結ぶ．

2 関連研究

本章では，はじめに既存のシナリオを用いた UI 設計について詳細を述べ，課題を明確化するとともに，いくつかの関連研究と技術について述べる．

2.1 シナリオを用いた UI 設計

シナリオを用いた UI 設計プロセスの多くは，図 2 に示した，1) フィールド調査，2) シナリオ作成と詳細化，3) スケッチ・プロトタイプの作成，4) 評価の 4 プロセスを含む構成となっており，これらを反復しながら設計を進める．以降，それぞれについて説明を述べる．

フィールド調査 フィールド調査では，対象ユーザが既存の枠組みの中で，どのように作業をしているかを観察やインタビューなどを通じて明らかにし，解決すべき課題や，上手く工夫している点などを発見する．これらは，システムを利用するユーザ像の明確化とシナリオ作成のための元データとなる．

シナリオ作成・詳細化 フィールド調査結果をもとに，ペルソナなどを作成して対象とするユーザ像を明確化するとともに，その対象ユーザが目的を達成するために，システムをどのような状況で，どのように利用するかなどを，シナリオとして記述する．フィールド調査から妥当なシナリオを記述する方法については，ゴールダイレクテッドデザイン [3] などで研究されている．

この段階では，具体的な機能や UI 要素は含まないように記述する．シナリオは専門的な知識がなくても容易に理解できるので，異なる立場の関係者（プログラマ，顧客など）とも合意形成を取りやすい．

機能や UI 要素を含まないシナリオを作成した後，シナリオからユーザの要求読み取り，さらに，ユーザの要求を実現するために必要な機能を抽出する．その後，機能を具体的にどのように使うのかをシナリオに書き加えることで，シナリオを詳細化する．

スケッチ・プロトタイプ作成 スケッチ・プロトタイプ作成では，詳細化したシナリオからどのような機能を一緒に使うか，どの順番で使うか，などを読み取り，画面遷移と各画面の大まかな機能の配置を検討する．さらに，各画面のユーザが機能を使うために必要な UI コンポネントを詳細化する．なお，この工程はシナリオの詳細化と共に行うことで，シナリオだけでは気がつかない新たな機能や操作を発見できる．さらにそれらを，シナリオでどのように記述するかを検討することで，機能や操作を付け加えることが妥当かを確認できる．これにより，機能の盛り込みすぎを予防できる．

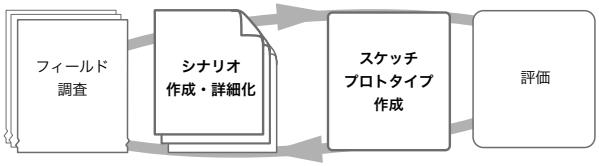

図 2. シナリオを用いる基本 UI 設計プロセス

評価 スケッチが完成した後，設計が妥当かを確認するために，ペーパプロトタイプ [10] などを用いてユーザビリティ評価を実施する．その後，評価結果をもとに，さらにシナリオとスケッチの修正を行う．

2.1.1 既存手法の課題

1 章で述べた通り，シナリオから UI を設計する際の難しさは「シナリオ作成・詳細化」と「スケッチ・プロトタイプ作成」のプロセスが乖離していることに起因する．そのため，シナリオを分析をしやすくするために，シナリオを詳細化する手順を 3 段階で規定した構造化シナリオ法 [13] や，ワークフロー図などを用いてシナリオ同士の共通点やトレードオフを分析する手法 [15] が提案されている．これらにより，要求の明確化やシナリオ同士の関連を分析できるようになる．しかし，シナリオの分析や，分析結果と UI との対応関係を維持する作業は設計者に任せられており，手間がかかる．

2.2 関係性維持の支援

要求工学の分野において，要求追跡という仕様書とプログラムとの関係を維持する研究がある [12]．元々は，ソフトウェア開発において，仕様変更にともなうコード修正範囲を正確に予測することが目的である．近年では，UML[1] や BPML[2] などのモデリング言語を用いるモデル駆動型設計において，モデルから UI を自動生成できるようにすることで，UI とモデルとの関係性の維持を支援する手法も提案されている [11]．これにより，仕様変更に際する，ユーザインタラクションや UI への影響範囲を予測できると考えられている．

モデルには，記述方法に厳密な定義があるため，定義に従って正確にかつ詳細にユーザ行動を記述することで，UI の自動生成が可能となる．しかし，定義の理解およびモデルの記述には専門知識を要する

[1] Unified Modeling Language
[2] Business Process Modeling Language

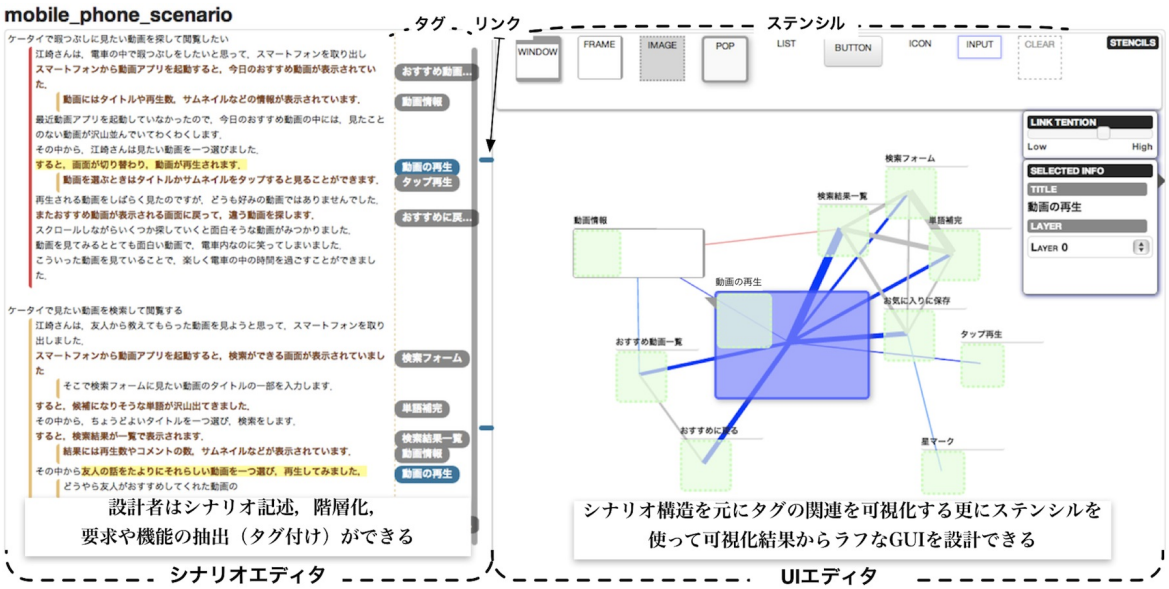

図 3. シナリオ順応型 UI 設計ツール

ことに加えて，厳密な定義のために表現の自由度が低下する．よって，専門性と正確性が重視される開発工程には適しているが，それらの専門知識を持たない，顧客やデザイナの理解を得ながら進める UI 設計の工程には適さない [14].

また，シナリオに基づく設計においても，シナリオと要求，プログラムとの関係性の維持を支援するシナリオブラウザ [9] や，シナリオを用いたアジャイル開発を支援する手法 [8] などが提案されている．しかしこれらは，開発においてユーザ要求と実製品とが乖離しないことを主眼としており，UI 設計工程において，UI との関係性の維持や設計に関しての支援は特になされていない．

2.3 プロトタイピングツール

UI 設計を反復するためには作業の効率化が重要であるため，主に GUI 設計において，Axure[1] や Expression Blend[6] などのプロトタイピングツールが数多く提供されている．プロトタイピングツールでは，あらかじめ用意された GUI コンポネントをドラッグ＆ドロップで配置することで，外観を素早く作成できる．また，クリックで画面を遷移させるなど，ある程度の操作が可能なプロトタイプも実現できる．さらに，ユーザビリティ評価までを支援するプロトタイピングツールも提案されている [5].

一方で，設計者は複数の利用シナリオにおける共通点やトレードオフを分析しながら UI を設計をする必要があるものの，プロトタイピングツールでは，それらの分析結果を参照，活用するための支援は不十分である．

本研究の主眼は，上記のプロトタイプツールを利用する「スケッチ・プロトタイプ作成」と，その前工程である「シナリオ作成・詳細化」とをスムーズに繋ぐことにある．よって，必要に応じて本研究とプロトタイピングツールとを組み合わせて，GUI を設計できることが望ましい．

3 シナリオ順応型 UI 設計ツール

シナリオ順応型 UI 設計ツールは，2.1 節で述べた「シナリオ作成・詳細化」および「スケッチ・プロトタイプ作成」の工程を支援する．第一に，シナリオを階層化することで，ユーザ要求を明確化し，それらのユーザ要求を満たすための機能を抽出しやすくする．第二に，シナリオの分析結果を GUI 設計に活かしやすくするために，分析結果を自動的に可視化して見やすくする機能を提供する．これらの機能により，UI 設計の専門家でなくても UI 設計の反復が容易になり，ユーザの要求を的確に捉えた，使いやすい GUI を設計しやすくする．

図 3 にツールの全体像を示す．ツールは大別してシナリオエディタと UI エディタに分かれており，2 種類のエディタを作業に合わせて使い分けながら GUI を設計する．以降，それぞれのエディタと支援機能の詳細について述べる．

3.1 シナリオの階層化とタグ付けの支援

設計者はシナリオを分析するために，シナリオエディタを用いて，シナリオの作成，シナリオの階層化，および機能の抽出を行う．

まず初めに，設計者はフィールド調査などで得られた実データをもとに，ユーザがどのようにシステムを利用するのかをシナリオとして記述する．シナリオは単純な操作の箇条書きではなく，ユーザの利用背景や，モチベーション，ゴールなどを含めて文章として記述する．

次に，設計者はシナリオの文毎に階層を定める．

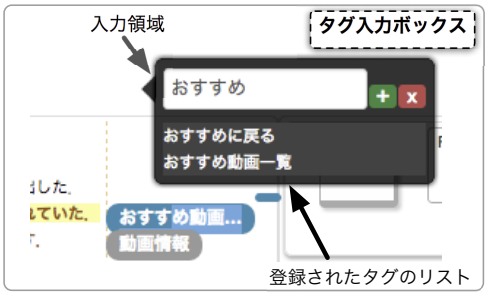

図 4. タグ入力ボックスを用いたタグの付与

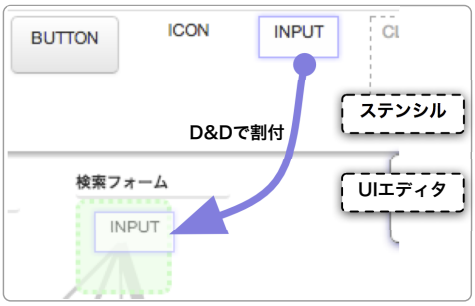

図 5. ノードへの GUI コンポネントの割り付け

図 3 左に階層化したシナリオの例を示す．設計者はシナリオエディタを用いて，第一階層はシナリオタイトル，第二階層は具体的な機能や GUI コンポネント（ボタン，ウィンドウなど）を含めずに要求だけを記述する詳細度が低いシナリオ，第三階層以降は，機能や GUI コンポネントが具体的に記述する詳細度が高いシナリオ，と階層を切り分けて記述する．これにより，詳しく記述された部分や記述が不足している部分など，詳細度のバラツキを容易に把握できるようになる．なお，本ツールは階層に合わせて記述に関するツールチップを表示するので，ツールチップに従って階層を調整することで，適切な階層を作りやすくなる．例えば，具体的な GUI コンポネントの表現が，第二階層で記述されている場合「具体的すぎるので第三階層に記述しましょう」などと促される．

最後に，設計者はシナリオ内の機能に該当する部分にタグを付与する．タグとは，シナリオ内の任意の文に付与できるテキスト情報である．本ツールはタグと文との関連を自動的に保持するので，機能（タグ）をどの文から抽出したのかを素早く確認できる．タグの付与は，対象のシナリオ内の文を選択した状態で，図 4 に示す「タグ入力ボックス」を利用して行う．例えば，図 1 の文からは「おすすめ動画一覧」といったタグが考えられる．なお，一度付けたタグは，タグ入力ボックスの下部にリスト表示されるため，再利用が容易である．また，タグのリストは入力領域でインクリメンタルサーチが可能であり，タグが増えても高速な選択が可能である．これらの機能により，設計者はタグを用いた機能の抽出と維持が効率的にできるようになる．

3.2 タグを用いたシナリオ可視化と UI 設計支援

シナリオを階層化した後，設計者はタグを用いて GUI を設計する．その際，GUI 設計を素早く行えるように，UI エディタでは自動的にタグ同士の関連を可視化する．これにより，設計者はタグ同士の関連を理解しやすくなる．図 3 右に可視化例を示す．可視化にはグラフ表現を用いており，タグはノード（四角＋破線枠）として，タグ同士の関連はエッジ（実線）として表現される．ノードに関しては，対象のタグを付与した回数が多いほど，グラフ上で大きく表現される．また，ノードを表示する位置はバネモデルで自動計算される．更に，シナリオ内で距離が近い（階層が同じで隣接している）タグのノード同士は，エッジで繋がれて近くに配置される．なお，グラフはエッジが増えると視認性が下がるため，エッジをどの程度表示するかは設計者が調整できるようにした．

タグ同士の関連をグラフを用いて可視化することで，「エッジが多いノードには複数のシナリオが関係しているので困難を招きやすい部分である」などと，注意すべきポイントを視覚的に把握しやすくなる．例えば，図 3 右の「動画の再生」というノードに注目すると，エッジが多くグラフの中心にあることがわかる．つまり，複数のシナリオに関係する中心的なノード（タグ）であると素早く理解でき，どのシーンで使うときも，混乱が起きないように UI を設計しなければならないと判断できる．そこで「動画の再生」に関する操作は，独立した画面を用意して他の操作と切り分けたほうが良い，などと検討できる．

さらに設計者は，グラフを直接操作してラフな GUI を設計できる．図 5 に示すように，ステンシルから GUI コンポネントを 1 つ選んでノードにドラッグ＆ドロップすることで，GUI コンポネントをノードに割り付けられる．さらに，自由にリサイズしてレイアウトすることが可能であり，最終的には図 6 左に示すような GUI を設計できる．さらにプロトタイピングツールを用いることで，例えば図 6 右のように詳細化できる．

なお，グラフは GUI コンポネントを割り付けた後でも重畳表示されるので，設計者は GUI コンポネントとグラフを同時に確認できる．また，グラフを非表示にして，GUI のみを表示することもできるので，作業の状況に合わせて表示を選べる．

3.3 シナリオと GUI とのトレーサビリティの提供

設計者がシナリオと GUI との間にある因果関係を素早く確認できるように，本ツールでは作成されたシナリオ，タグ，ノードの対応関係を自動的に保持する．これにより，GUI 設計を反復するうちに，

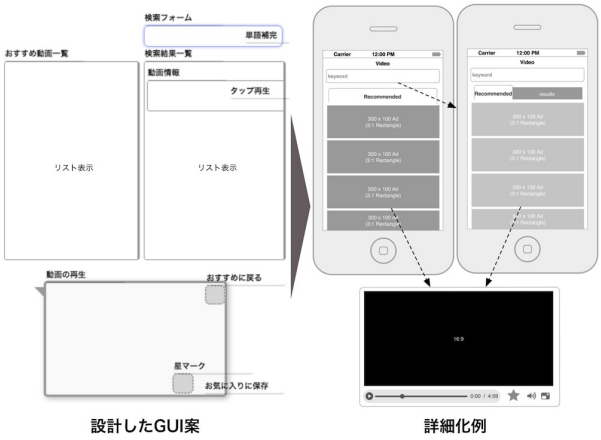

図 6. 設計した GUI 案と GUI 案の詳細化例

シナリオと GUI とが乖離してしまうことを防ぐ．

本ツールでは，設計者がシナリオ，タグ，ノードのいずれかを選択すると，図 7 に示すように，関連しているシナリオ，タグ，ノードを自動的にハイライトする．ハイライトされると色が変化して濃く表示される．加えて設計者は，選択中のタグを，シナリオ全体のどこに何箇所あるのかを,「リンク」によって素早く把握できる．リンクとは，スクロールバーの右横に表示される小さなアイコンのことで（図 7 中央）．これにより，スクロール操作をしなくても，シナリオの何処に選択中のタグがあるかを一覧できる．このような，トレーサビリティとインタラクションを提供することで，参照に対する手間を軽減し，設計者はシナリオの分析と GUI の設計に集中しやすくなる．

さらに，設計者は GUI をユーザテストする際に，本ツールを用いることで，GUI が想定するシナリオを素早く参照できるので，テスト用のシナリオを生成しやすくなる．特に，シナリオは階層化されているため,「大ざっぱな利用シーンだけを提示して自由に操作する様子を観察する」または「ユーザの振る舞いを含めたシナリオを提示して認知的ウォークスルーを実施する」など，評価したい内容に合わせて，詳細度を変化させたシナリオを効率良く生成できる．これらの支援により，GUI 設計と評価の反復を効率化できる．

4 実装

本システムは JavaScript と HTML5 を用いて実装されており，Google Chrome[3] 上であれば，OS に非依存で動作する．

シナリオエディタは DIV 要素の contenteditable 属性を true に設定することで，部分的な編集権限をユーザに与えた．加えて，キーボードイベントを拡張し，Tab キーによる階層付け操作などのショートカットを実現した．

[3] http://www.google.com/chrome

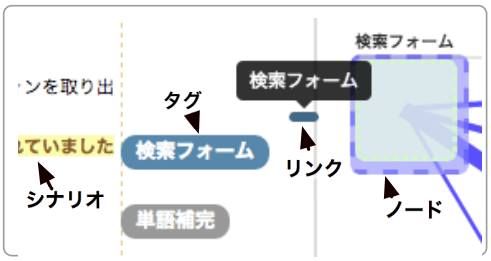

図 7. シナリオ，タグ，ノード (UI) のハイライト

UI エディタでは，ノードを DIV 要素で表現し，エッジは Canvas 上に KineticJS[4] を用いて描画した．GUI コンポネントの割り付けは，ステンシルとノードのドラッグ＆ドロップイベントを用いて，ノードの CSS をステンシルの CSS に書き換えることで実現している．なお，UI エディタのマウス操作に関しては JQuery UI[5] を用いて実現した．

データを保存する際には，UI とシナリオのデータをそれぞれ XML 情報に変換して保存する．保存場所については Localstorage を利用した．

5 ユーザスタディ

実装したツールを用いてユーザスタディを実施した．特に，シナリオ階層化，タグ付け，グラフを用いた UI 設計の操作に着目して，どのように利用されるかを実際に観察することで，提案した機能が意図した通りに使われるかを検証した．

5.1 条件

ユーザスタディでは，スマートフォンで利用する動画アプリに関して，シナリオの作成と，そのシナリオに適した GUI の設計を依頼した．実施前にシナリオ作成の目的，作業の確認，および本ツールの操作方法を説明した．想定シーンとしては「暇つぶしに見てみたい動画を探して閲覧するシーン」と「目的の動画を検索して閲覧するシーン」の 2 つを与えた．参加者はシナリオに基づく UI 設計の経験がない 3 名の研究者で，全員がスマートフォンと動画アプリの利用者であった．作業中は観察者が作業を観察するとともに，カメラ撮影と操作画面の録画とを行った．さらに作業後，画面録画を見ながら，操作感や戸惑った点などについてインタビューを行った．

5.2 結果と考察

シナリオ階層化に関して，階層化すると記述が不足している部分が即座に理解できる，というコメントが得られた．これは，階層化によって狙い通りの気付きを参加者に与えられたと考えられる．一方で，最初から具体的な操作を箇条書きしたくなる，階層毎にシナリオの詳細度を揃えることが難しい，といっ

[4] http://www.kineticjs.com/
[5] http://jqueryui.com/

たコメントも得られた．このことから，シナリオ記述の未経験者が使う場合には，シナリオの記述に関しても，具体的なサンプルを表示する，チュートリアルを用意するなどの支援が必要であると考えられる．

タグの付与とタグ同士の関連の可視化に関しては，システムで中心的なタグがどれなのかを素早く把握できるとのコメントが得られた．シンプルなグラフによる可視化だが，ノード（タグ）同士の依存関係の把握に効果があることが示唆された．しかし，全体としてみると，グラフよりもシナリオを確認する方が好まれた．これは，記述したシナリオの規模が小さく，全体像が把握しやすかったためだと考えられる．

UI の設計に関して，ステンシルを用いて GUI を設計しているときに，GUI の操作に必要な機能が足りないことに気がつく様子が観察された．その際に，操作に関係するシナリオ内の文を素早く参照して，文の追記と新たなタグの付与をしていた．これは「シナリオ詳細化」と「スケッチ・プロトタイプ作成」の工程が交互に行われたといえる．一方で，「戻る」や「閉じる」ボタンなどの，細かい GUI コンポネントを作り込もうとすると，それらに対応する文をシナリオ内に記述する必要があり，煩雑であるというコメントが挙げられた．このことから，さらに実用性を高めるためには，詳細な GUI を作る際に，文を追記しなくてもシナリオと対応付けできるようにするなど，作業の工程にあわせた操作性の改善が必要であると考えられる．

上記の結果から，本ツールはシナリオの分析，および GUI 設計に必要な情報の参照を効率化することが示唆された．加えて，ツールの実用性を高めるために，さらなるサポートが必要な操作を洗い出すことができた．

6 おわりに

本稿では，シナリオを用いた GUI の設計と評価の反復を的確に，かつ効率的にできる，シナリオ順応型 UI 設計ツールについて紹介した．具体的には，シナリオの階層化による要求の明確化と，タグの付与による機能の抽出を支援する方法，およびタグ同士の関連を可視化することによって UI 設計を支援する方法について紹介し，その効果を述べた．

今後は，実際の UI 改善事例などに積極的に本ツールを導入し，より詳細に本ツールの効果を検証していく必要がある．既に実現している機能の使い勝手をさらに向上させることはもちろんのこと，ツール上で設計した GUI を操作可能なプロトタイプとしてエクスポートできるようにするなど，UI 設計の更なる効率化に取り組んでいきたい．

参考文献

[1] Axure. Axure, 2012. http://www.axure.com/.

[2] J. M. Carroll. *Making Use: Scenario-Based Design of Human-Computer Interactions*. MIT Press, 2000.

[3] A. Cooper, R. Reimann, and D. Cronin. *About face 3: the essentials of interaction design*. John Wiley & Sons, Inc., New York, NY, USA, 2007.

[4] ISO. ISO 9241-210 Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. Directly by ISO, 2010.

[5] Y. Li and J. A. Landay. Activity-based prototyping of ubicomp applications for long-lived, everyday human activities. In *Proc. CHI '08*, pp. 1303–1312, New York, NY, USA, 2008. ACM Press.

[6] Microsoft. Expression Blend, 2012. http://www.microsoft.com/expression/.

[7] J. Nielsen. Iterative user-interface design. *Computer*, 26(11):32–41, 1993.

[8] H. Obendorf and M. Finck. Scenario-based usability engineering techniques in agile development processes. In *Proc. CHI EA '08*, pp. 2159–2166, New York, NY, USA, 2008. ACM Press.

[9] M. B. Rosson and J. M. Carroll. Integrating task and software development for object-oriented applications. In *Proc. CHI '95*, pp. 377–384, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[10] C. Snyder. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann, 2003.

[11] K. Sousa, H. Mendonca, J. Vanderdonckt, E. Rogier, and J. Vandermeulen. User interface derivation from business processes: a model-driven approach for organizational engineering. In *Proc. SAC '08*, pp. 553–560, New York, NY, USA, 2008. ACM Press.

[12] S. Winkler and J. von Pilgrim. A survey of traceability in requirements engineering and model-driven development. *Software and Systems Modeling*, 9:529–565, 2010.

[13] K. Yanagida, Y. Ueda, K. Go, K. Takahashi, S. Hayakawa, and K. Yamazaki. Structured Scenario-Based Design Method. In M. Kurosu ed., *Human Centered Design*, Vol. 5619 of *Lecture Notes in Computer Science*, pp. 374–380. Springer, 2009.

[14] 妻木 俊彦, 白銀 純子, 本位田 真一. 要求工学概論 - 要求工学の基本概念から応用まで. 近代科学社, 2009.

[15] 棚橋弘季. ペルソナ作って、それからどうするの? ユーザー中心デザインで作る Web サイト. ソフトバンククリエイティブ, 2008.