

UI-Filler：シナリオに基づく対話型UI設計支援ツール

草野 孔希^{1,a)} 中谷 桃子^{1,b)} 大野 健彦^{1,c)}

受付日 2013年5月17日, 採録日 2013年11月1日

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, UI-Filler 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, ユーザの振舞いを熟慮しながら繰り返し UI を設計することが重要である. そこで, 振舞いを物語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, 従来のシナリオに基づく設計手法には, 1. 複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, シナリオどうし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, 2. 設計した UI を評価しながら繰り返し改善する際に, シナリオと UI との対応関係を維持することが難しい, という2つの難点がある. そこで, UI-Filler 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と管理の支援, および分析結果の可視化によって UI の反復設計を支援する.

キーワード：インタフェースデザイン, グラフィカルユーザインタフェース, シナリオ, デザインツール

UI-Filler: Interactive Design Tool for Scenarios and UIs

KOKI KUSANO^{1,a)} MOMOKO NAKATANI^{1,b)} TAKEHIKO OHNO^{1,c)}

Received: May 17, 2013, Accepted: November 1, 2013

Abstract: Clearly picturing user behavior is one of the key requirements when designing successful interactive software. However, covering all possible user behaviors with one UI is a complex challenge. UI-Filler is designed to support the characterization of user behavior based on scenarios and then using the information in UI design. Scenarios make it easy to understand and share user behavior even if we have little design knowledge. However, they have two big weaknesses; 1) integrating several scenarios in one UI is difficult, even if we can create appropriate scenarios, 2) maintaining the links between scenarios and the UI is a heavy task in iterative design. Our tool solves the above problems through its hierarchical scenario structure and visualized overview of scenarios. It enhances the designer's skill in writing scenarios and designing UIs smoothly and easily.

Keywords: interface design, graphical user interface, scenario, design tool

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舞いや環境を

熟慮して UI を設計することが重要である [1]. このための方法として, シナリオを作成することは, ユーザの振舞いを具体化し, ユーザ要求を明確にするうえで有効であることが Human-Computer Interaction (HCI) 分野において広く知られている [2], [3]. シナリオとは, 1人以上のユーザが製品を使って特定の目的を達成するまでの様子を物語として記述したものであり [3], 図 1 に示すとおり自然言語を用いて記述する. 操作手順を列挙するだけではなく, ユーザの目的や状況も含めて物語調に記述することがシナリオの特徴である. これにより, 設計者や立場の異なる関係者 (プログラマ, 顧客など) がユーザの具体的な利用シー

¹ NTT サービスエボリューション研究所
NTT Service Evolution Laboratories, Yokosuka, Kanagawa
239-0847, Japan

a) kusano.kouki@lab.ntt.co.jp

b) nakatani.momoko@lab.ntt.co.jp

c) ohno.takehiko@lab.ntt.co.jp

現場作業終了後に、中野は帰りのワゴン車内で作業者にヒアリングをする。まず、作業者に現地ですべては何をするかを聞いたところ「お客様のところに行く前にまずは屋外装置の下調べする。下調べをしておくとお客様の印象として短時間で作業しているように見える」と答えた。中野はこれを大切だと思ったので、メモに「ノウハウ：事前に下調べをして、お客様の待時間を減らす」と記録した……

図 1 シナリオの記述例

Fig. 1 Example of scenario.

ンを想像しやすくする。

一方でシナリオには、シナリオ数の増加にともない、シナリオの分析や更新が面倒になるという問題がある [4]。これは、各シナリオの読解に加えて、シナリオどうしの共通点とトレードオフの分析に手間がかかるためである。加えて、使いやすい UI を実現するには、設計と評価を反復しながら UI を改善することが重要である [5], [6]。しかし、分析の手間がかかることで、UI 改善時の円滑な反復が阻害されてしまう。

そこで本研究では、UI 設計におけるシナリオ分析や更新の時間の軽減するツール、UI-Filler を実現する。UI-Filler を利用することで、シナリオを用いた UI 設計の専門家でなくても、シナリオを用いてユーザの要求を的確にとらえながら効率良く UI 設計と評価を反復できるようにし、使いやすい UI を設計しやすくする。

以降、2 章では関連研究や技術について紹介し、3 章で UI-Filler が満たすべき 3 要点を定義する。4 章では UI-Filler の機能を具体的に紹介し、5 章で実装について述べる。さらに、UI-Filler の有効性について 6 章で議論し、最後に 7 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、既存のシナリオを用いた UI 設計のプロセスについて述べ、課題を明確化するとともに、関連分野の研究と技術を紹介する。

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 に示すとおり、1) フィールド調査、2) シナリオ作成、3) UI 設計、4) UI 評価の工程を含む構成であり、これらを反復しながら設計を進める。以降、各工程について簡単に述べる。

1) フィールド調査

フィールド調査では、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、うまく工夫している点などを発見する。

2) シナリオ作成

調査結果をもとに、ペルソナ [7] などを作成して対象のユーザ像を明確化し、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、どのように利用したかを含めて、シナリオを記述する [3]。シナリオはユーザの利用

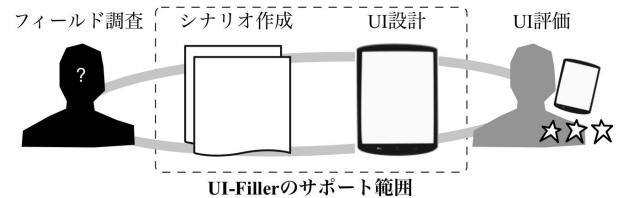


図 2 基本的なシナリオ設計プロセス

Fig. 2 Basic scenario-based design process.

シーンごとに分けて作成する。設計の初期段階で記述するシナリオは、設計の選択肢を狭めないように、具体的に記述しすぎないことが重要である。そのため、まずは具体的な要件や UI 表現 (e.g. ボタン、入力フォーム、ウィンドウ) は含まないようにする。

具体的な UI 表現を含まないシナリオを作成したうえで、それらのシナリオからシステムに求められる要件を抽出する。ここでの要件とは、UI の設計の初期段階において必要な、ユーザが UI の中で使う機能 (e.g. メモをする機能) と、その機能がアクセスするデータ表現 (e.g. メモ、録音データ) を指す [3]。次に、抽出した要件をどのような UI コンポーネントで実現し、ユーザがどのような順番で操作するのかなど、実現性を加味しながら具体的なシナリオを別途作成する。

3) UI 設計

UI 設計では、要件と UI コンポーネントを具体化したシナリオ群から、どのような要件を一緒に使うか、どの順番で使うか、他シナリオとの共通点とトレードオフなどを分析しながら、画面遷移と各画面の UI コンポーネントの配置を検討する。なお、UI 設計中にシナリオで検討漏れをしていた UI コンポーネントや要件に気がついた場合には、シナリオでそれらがどのように使われるのかを検討し、UI コンポーネントや要件を付け加えることが妥当かを確認する。この検討により要件の盛り込みすぎの予防をできる。このように、UI 設計とシナリオ作成を合わせて行うことで、過不足ないシナリオと UI を設計しやすくなる。

4) UI 評価

UI を設計したあとで、設計者はシナリオと照らし合わせながら UI を評価する。また、ペーパープロトタイプ [8] などで UI のユーザ評価する。さらに、必要に応じて一部機能が実際に使えるプロトタイプを作成し、応答速度なども含めた操作性を評価検討する。その後、評価結果をシナリオと UI に照らし合わせながら、シナリオと UI を改善する。

2.1.1 既存手法の課題

シナリオはユーザの振舞いを明確化し、システムに求められる要件を抽出する際に有効であるが、シナリオの数が増えると扱いが難しくなる [4]。具体的には、抽出される要件が増えることに加えて、シナリオどうしの共通点とトレードオフの分析が難しくなることがあげられる。分析で

は各シナリオを理解する手間は当然として、すべてのシナリオを俯瞰することにも手間がかかる。これらの手間を少なくし、シナリオをより効果的に使うために、シナリオの全体像を把握できる分かりやすい表現が求められている [9]。

そこで、シナリオを整理するために、シナリオの詳細化手順を 3 段階で規定した構造化シナリオ法 [10] や、フローチャートなどを用いてシナリオどうしの共通点とトレードオフを分析する手法 [11] が提案されている。これらにより分析の正確性は増すものの、各資料の作成や資料どうしの対応関係を維持する手間が増える。加えて、使いやすい UI を設計するには評価と改善を反復することが重要であるが、維持管理の手間によって円滑な反復を阻害されてしまう。

2.2 対応関係維持の支援

要求工学において、要求追跡という仕様書とプログラムとの対応関係を維持する研究がされている [12]。元々は、ソフトウェア開発において、仕様変更にもなうコードの修正範囲を正確に予測することが目的である。加えて、BPML^{*1}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 を自動生成することで、UI とモデルとの対応関係の維持を支援する手法も提案されている [13]。これにより、仕様変更にもなう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いわれている。

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システムの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ーの理解を得ながら進める必要がある UI 設計には不向きである [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、プログラムとの対応関係の維持を支援するシナリオブラウザ [15] や、シナリオを用いたアジャイル開発を支援する手法 [16] 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装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 とシナリオとの対応関係の維持や UI 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。そのため、主に GUI 設計においてプロトタイピングツールが数多く提供されている [17]。プロトタイピングツールを用いると、用意された GUI コンポーネントをドラッグ&ドロップして配置するだけで、外観を素早く作成できる。これらの中には、クリックで画面を遷移させるなどの操作が

可能なプロトタイプを作成できるものもある [18], [19]。さらに、ユーザビリティ評価までを支援するツールも提案されている [20]。一方で、設計者は複数の利用シナリオにおける共通点とトレードオフを分析しながら UI を設計する必要があるものの、プロトタイピングツールでは、それらの検討項目を参照、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「UI 設計」と、その前工程である「シナリオ作成」とをスムーズにつなぐことにある。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る GUI を設計することが望ましい。

3. 支援すべき 3 要点

既存研究や関連研究をもとに、UI-Filler が支援すべき 3 要点を下記のとおり定める。これらを満たすことで、シナリオどうしの関係性やトレードオフの分析、およびシナリオと UI との対応関係の維持を省力化し、シナリオを用いた UI の反復設計を効果的かつ効率的に行えるようにする。

A. シナリオと要件の管理

作成したシナリオ、およびシナリオから抽出した要件の管理をしやすいこと。本ツールでは、詳細度の異なるシナリオの管理を容易にするために、アウトラインプロセスのようにシナリオ内の文に階層を付けられる機能を提供し、シナリオを別個に作らなくても詳細度を適切に管理できるようにする。さらに、シナリオから抽出した要件が、シナリオのどこから抽出されたものなのかを後からたどれる機能を提供する。

B. 複数シナリオの全体像の把握

複数あるシナリオの全体像を素早く把握できるようにすること。本ツールでは、抽出された要件とシナリオの構造をもとに、複数のシナリオからなるシステムの全体像を自動的に可視化する機能を提供する。

C. シナリオと UI との対応関係の維持

ユーザ評価結果から UI やシナリオを修正する際に、シナリオと UI を素早く相互に参照できるようにすること。本ツールでは、対応付けられたシナリオと UI との対応関係を自動的に維持し、設計者の操作に応じて素早く対応関係を提示する機能を提供する。

4. UI-Filler

UI-Filler は、3 章で述べた 3 要点を満たす機能を提供することで、2.1 節で述べたプロセスの「シナリオ作成」および「UI 設計」の工程を結び、UI の反復設計 [5], [6] をシナリオを用いて円滑に行えるようにすることで、使いやすい UI の実現を支援するツールである。シナリオを用いた

^{*1} Business Process Modeling Language

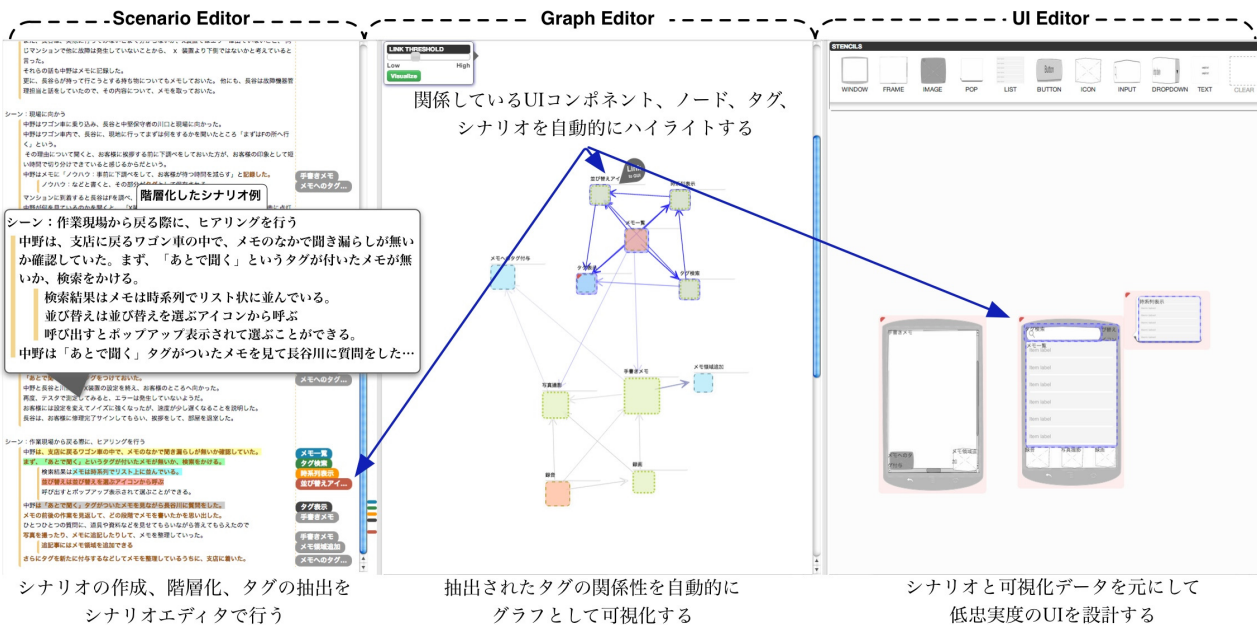


図 3 UI-Filler 外観

Fig. 3 The GUI of UI-Filler.

UI 設計の専門家であっても効率的に作業できることを目指している。特に、本ツールは PC などで作動作するインタラクティブシステムの GUI 設計に携わる設計者の支援を主眼に機能を実現した。

図 3 にツールの全体像を示す。ツールは Scenario Editor, Graph Editor, UI Editor の 3 領域に分かれている。本ツールを利用する設計者は、Scenario Editor でシナリオの作成と要件の明確化を行い、次に Graph Editor においてシナリオを分析し、その分析結果をもとに UI Editor で UI を具体化して評価する、という 3 つの工程を繰り返しながら UI の設計を進める。以降、UI-Filler がどのような機能を提供することで、上述の工程を支援するのか詳述する。

4.1 シナリオと要件の管理の支援

まずはじめに、設計者はシナリオ分析のために、Scenario Editor を用いて、シナリオの作成と階層化、および要件抽出を行う。

シナリオの作成では、はじめに高い抽象度のシナリオを記述し、その後で具体的な GUI コンポーネントなどを含むシナリオを追記する。その際に設計者は、第 1 階層はシナリオのタイトル、第 2 階層は具体的な GUI コンポーネント (e.g. ボタン、ウィンドウ) を含めずにユーザの振舞いだけを記述する抽象度が高いシナリオ、第 3 階層以降は、GUI コンポーネントの振舞いを具体的に記述する抽象度が低いシナリオ、と階層を分けて記述する (図 3 左参照)。階層化により抽象度の高いシナリオから、低いシナリオまでを 1 つの資料として保持できる。さらに、詳しく記述された部分や記述が不足している部分など、抽象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



図 4 タグ入力ボックスを用いたタグの付与

Fig. 4 Floating tag box.

わせて記述に関するツールチップを表示することで、各文の階層の調整を支援する。これにより、シナリオに馴染みがない設計者でも、階層を分けたシナリオの記述をしやすくする。

次に設計者は、シナリオから要件 (ユーザが UI の中で使う機能およびデータ表現) と UI に関する記述を見つけてタグを付与する。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。本ツールは付与したタグと文との対応関係を自動的に保持するので、タグをどの文に付与したのかを素早く確認できる。タグの付与には図 4 に示す「タグ入力ボックス」を用いる。なお、付与したタグは、タグ入力ボックスの下部にリスト表示されるため、再利用が容易で表記揺れを防げる。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できるので、タグが増えても検索が容易である。これらの機能により、タグを用いた要件の抽出と維持を容易にする。

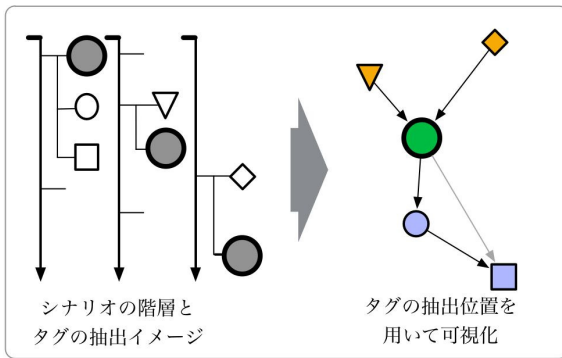


図 5 シナリオ内のタグの位置からグラフを自動作成

Fig. 5 Visualization process based on scenarios and tags.

4.2 複数シナリオの全体像把握の支援

設計者が Scenario Editor でシナリオを作成し、タグを抽出することで、Graph Editor においてタグどうしの関連が自動的に可視化される。これにより、設計者は複数のシナリオから抽出されたタグどうしの関連を理解しやすくなる。図 3 中央に可視化例を示す。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、タグどうしの関連はエッジ（実線）として表現される。

ノードは、対応するタグの抽出回数が多いほどグラフ上で大きく表現される。これにより、グラフにおいて抽出回数が多いタグが注目されやすくなる。また、出次数が多いノードは橙色、入次数が多いノードは水色、その他のノードは緑色で表現される。この色分けにより、シナリオにおける大まかなタグの出現順序を把握できる。なお、ノードの表示位置はばねモデル [21] を用いて自動計算される。

エッジは、図 5 に示すように、シナリオ内で距離が近い（階層が近くて隣接している）タグに対応するノードどうしの間に作成される。エッジの向きは、シナリオ中のタグ（ノード）の登場順で決まる。このようにグラフを用いてタグどうしの関連を可視化することで、「エッジが多いノード（タグ）は複数のシナリオで使われるので困難を招きやすい」などと、注目すべきタグを視覚的に把握しやすくなる。

設計者は自動的に作成されたグラフをもとに、UI Editor で GUI を設計する。図 6 に示すように、Graph Editor のグラフからノードを選択して UI Editor にドラッグ&ドロップすることで、ノードに対応付けられた「コンテナ」を作成できる。次に、コンテナに対してステンシル（図 6 上）と呼ばれる GUI コンポーネントの型を 1 つ選んでドラッグ&ドロップすると、GUI コンポーネントをコンテナに割り付けられる。なお、1 つのノード（タグ）が複数箇所でも利用されることも考慮し、1 つのノード（タグ）に対して複数のコンテナを割り当てられる。このようにして、図 8 中央に示すようにコンテナを組み合わせレイアウトし、最終的にステンシルを使って図 8 右に示すようなラフな GUI を設計する。

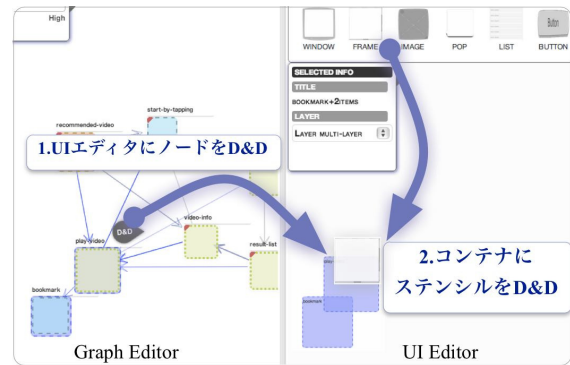


図 6 ノードと GUI コンポーネントとを対応付ける操作

Fig. 6 Linking steps of GUI components and nodes.

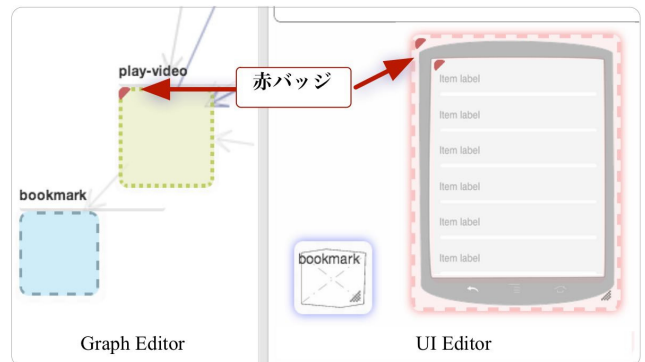


図 7 未対応であることを通知する赤バッジ

Fig. 7 Warning unlinked status between graph and UI.

著者らは、以前にシナリオの作成と GUI 設計を統合したツールを提案した [22]。このプロトタイプでは、GUI 設計時にグラフに対して UI コンポーネントを直接割り当てる操作を採用していた。しかし、ユーザテストの結果、この方式は設計した UI とグラフとを比較検討することが困難であり、さらに、グラフが複雑かつ大きくなると UI 設計時の操作性が低下するという課題が発見された。そこで、UI-Filler では、グラフを表示するエリア（Graph Editor）と UI を設計するエリア（UI Editor）とを分けた。これにより UI とグラフとを比較しやすくとともに、後述する対応関係維持の機能を提供することで、シナリオ、グラフ、UI の対応関係を簡単に参照できる環境を実現した。

4.3 シナリオと UI との対応関係維持の支援

UI-Filler はシナリオ、タグ、ノード、UI などの要素の対応関係を自動的に維持する。これにより、設計者は効率良く対応付けられた要素を参照できるようになり、シナリオを用いた UI の反復設計がしやすくなる。

本ツールでは、設計者が文、タグ、ノード、UI のいずれかの要素を選択すると、対応するすべての要素を自動的にハイライトする。ハイライトされた要素は色が変わって濃く表示される。さらに設計者は、選択中のタグが、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

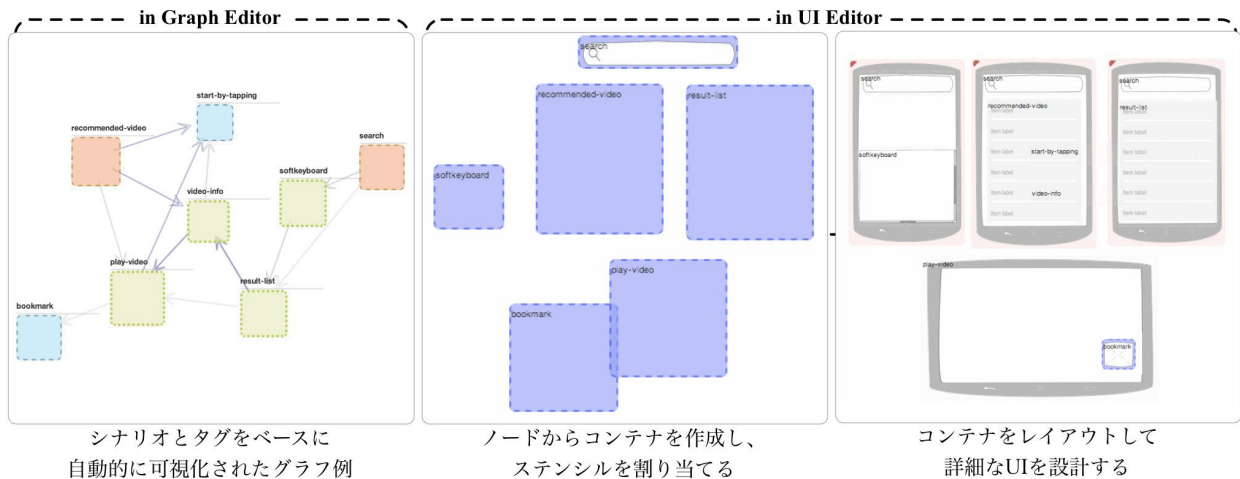


図 8 グラフをもとにして GUI を設計するためのステップ

Fig. 8 An example of designing UI steps through visualization.

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アイコンである。これによりスクロール操作をしなくても、選択中のタグの位置と数を一覧できる。さらに、図 7 に示すとおり、GUI コンポーネントと対応付けられていないノードには赤バッジを表示し、抽出されたノード（タグ）から GUI コンポーネントを設計し忘れることを防ぐ。

また、GUI とシナリオとの対応関係が明確になるため、GUI 評価用のテストシナリオを作成しやすくなる。たとえば、シナリオの階層を活かして「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「ユーザの振舞いを含めたシナリオを提示してステップごとに評価する」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作成できる。これらの支援により、設計者は GUI 設計と評価の反復を効率化できるようになり、使いやすい UI を実現しやすくなる。

5. 実装

UI-Filler は JavaScript と HTML5 で実装されており、Google Chrome^{*2}上で OS に非依存で動作する。Scenario Editor は DIV 要素の contenteditable 属性を利用することでテキスト入力を実現した。さらに、キーボードイベントを拡張し、階層付けの操作やタグ入力ボックスの呼び出しなどのショートカットを実装した。Graph Editor では、ノードを DIV 要素の CSS で描画し、エッジは KineticJS^{*3}を用いて描画するようにした。UI Editor における、コンテナへの GUI コンポーネントの割付けは、ドラッグ&ドロップイベントを用いて、コンテナの CSS をステンシルの CSS に書き換えることで実現した。なお、UI エディタのマウス操作に関しては JQuery UI^{*4}を用いて実現した。最後に、

データの保存に関しては、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 Chrome が管理する Localstorage を利用した。

6. ユーザスタディ

支援すべき 3 要点が、UIFilter が提供する機能（階層化シナリオとタグ管理、全体像の自動可視化、対応関係維持）によって適切に支援されているかを分析するために、3 種類の調査を実施した。第 1 に、本ツールを用いたシナリオの作成やタグ抽出の操作に着目し、ユーザが適切に機能を活用できるかを調査した。第 2 に、ユーザが本ツールを用いて GUI を設計する様子を観察し、GUI 設計において本ツールがどのように活用されるのかを分析した。第 3 に、グラフ表現による可視化について、グラフから UI 設計に必要な情報を読み取れるかを検証した。

6.1 調査 1：シナリオ作成とタグ付与作業の観察

UI-Filler を用いて、シナリオ作成およびタグ付与の機能をユーザが適切に利用できるかを調査するために、3 名の調査参加者にシナリオの作成作業を依頼した。

6.1.1 実施手順

まずはじめにシナリオとタグの意味について簡単に説明し、練習時間を 10 分程度設けて、その間はシナリオの書き方やタグの付与の仕方などについて質問があれば適宜答えた。練習後、2 つの想定シーン（小シナリオ）を含むシナリオを和文 1,000 字程度で作成し、タグを付与するように指示した。さらに、各作業終了後にインタビューを実施し、作業中に行った操作の意図や判断について質問した。なお、3 名の参加者は著者らの所属する組織の研究者（男性 3 名）で、UI 設計のためにシナリオを作成した経験はなかった。

6.1.2 実験結果と考察

まずはじめに、10 分程度の練習時間で全参加者が Sce-

^{*2} <http://www.google.com/chrome>

^{*3} <http://www.kineticjs.com/>

^{*4} <http://jqueryui.com/>

nario Editor を用いたシナリオの作成、およびタグの付与に関する操作はスムーズにできている様子が観察できた。加えて、シナリオを階層化することで、記述が不足している部分が明確になって良いというコメントが得られた。また、タグを付与することでシナリオを整理できるというコメントも得られた。一方で、最初に具体的な操作を箇条書きしたくなる、階層ごとの文の詳細度を揃えることが難しい、といったコメントも得られた。

以上の結果から、シナリオ作成とタグ付与に関する操作性は問題ないと考えられる。また、シナリオ自体は日頃から目にする馴染みのあるものであるため、説明を受ければシナリオの作成自体は問題なく行えると考えられる。しかし、より利用者がシナリオ作成や階層化を円滑にできるようにするには、チュートリアルを用意するなど、支援機能を提供することが望ましいと考えられる。

6.2 調査 2：GUI 設計作業の観察

UI-Filler が GUI の設計作業においてどのように利活用されるのかを調査するために、5 名の調査参加者に UI-Filler を用いた GUI 設計作業を依頼した。

6.2.1 実施手順

まずはじめに、シナリオとツールについて簡単に説明を行い、作業前に操作練習を 10 分程度設けた。練習中は操作やシナリオに関する質問に適宜答えた。練習後、シナリオを読んで GUI を設計するように依頼した。調査参加者には、全体像の可視化と対応関係の維持の機能がある場合とない場合とで、それぞれ UI をデザインするように依頼した。2 種類のシナリオを用意し、各条件でどちらのシナリオを使うかは参加者ごとに変更した。シナリオは和文 (2,800 文字、3,500 文字) で作成し、各シナリオは 5 つの異なる想定シーン (小シナリオ) で構成した。タグはあらかじめ実験者が抽出したものを扱うように指示した。タグは特定の小シナリオでのみ使う要件、小シナリオ共通で使う要件とがそれぞれ含まれた。また、各作業終了後にアンケートとインタビューを実施し、作業中に行った操作の意図や判断について質問した。なお、参加者は著者らの所属する組織の研究者 (男性 3 名、女性 2 名) で、シナリオを用いた UI 設計の経験はなかった。

6.2.2 調査結果と考察

シナリオと要件の管理について

階層化したシナリオは、従来のシナリオに比べて特殊な表現ではあったが、全員がシナリオから利用シーンをイメージできたと回答した。加えて、シナリオを読んだ後は、必要に応じて抽出された要件 (タグ) だけを主に参照し、タグが付与された文の前後を読むことで詳細を復習する様子が見られた。その理由として、シナリオは冗長な表現が多いため、シナリオを読み直すよりは、タグを中心に確認する方が効率が良い、というコメントが得られた。以上の

結果から、ユーザは短時間の説明を受けるだけでシナリオとタグの意味を理解し、かつタグとシナリオとを組み合わせることで、シナリオの理解を効率化できることが示唆された。

複数シナリオの全体像の可視化について

グラフありの場合、グラフをもとにタグどうしの関連を確認し、大まかな画面遷移を設計していく様子が観察できた。また、グラフでタグを確認したうえで、タグに関連するシナリオを、ハイライトされた情報をもとに参照する様子が見られた。加えて、グラフなしの場合は小シナリオごとに UI を設計していたのに対して、グラフありの場合はタグどうしの関連をもとに UI を設計していたとのコメントも得られた。さらに、ある小シナリオ用に UI を設計した後で、UI に用いたタグに関連するタグをグラフで確認し、関連するタグとシナリオを加味して UI を修正する、という評価と改善を繰り返す様子が見られた。

以上の結果から、自動生成されたグラフは、タグどうしの関連を俯瞰、つまりシナリオ間の関係性を加味しながら GUI を設計するうえで有効であることが示唆された。しかし、調査 2 によって設計者の設計と評価の作業を効率化できることは示唆されたものの、具体的にグラフのどのような特徴が全体像の把握に有効だったかが曖昧であったため、別途可視化に関する調査を実施した。詳細は 6.3 節で述べる。

シナリオと UI との対応関係の維持について

すべての調査参加者は、最初の操作説明と練習時間を通じて、GUI とシナリオとの対応付けの操作を行えるようになった。また、UI 設計において、対応関係が維持されることで、状況に応じて GUI から対応するシナリオを参照できている様子が観察された。さらに、UI 設計の後でグラフを確認し、ノードに表示された赤バッジを見て、GUI コンポーネントが未割当てのノードがあることに気がつき、UI を修正する様子が観察された。一方で、対応関係の維持がない場合、1 名が余分な GUI コンポーネントを配置したほか、1 名が 1 つのタグに対応する GUI コンポーネントの配置を忘れた。この結果から、設計した UI の確認と修正において、対応関係維持の機能の有効性が示唆された。なお、シナリオと GUI との対応関係維持の機能は UI の設計作業を容易にすると、すべての調査参加者が回答した。以上の結果から、UI 設計に関する操作性を損なうことなく、対応関係の維持を実現できたと考えられる。また、GUI に関連するグラフやシナリオの参照が容易になることで、シナリオを用いた UI 設計と評価の反復を効率化できることが示唆された。

6.3 調査 3：可視化の妥当性の検証

UI-Filler が生成するグラフが、複数シナリオの全体像を把握するうえで、どのように活用されるのかを調査した。

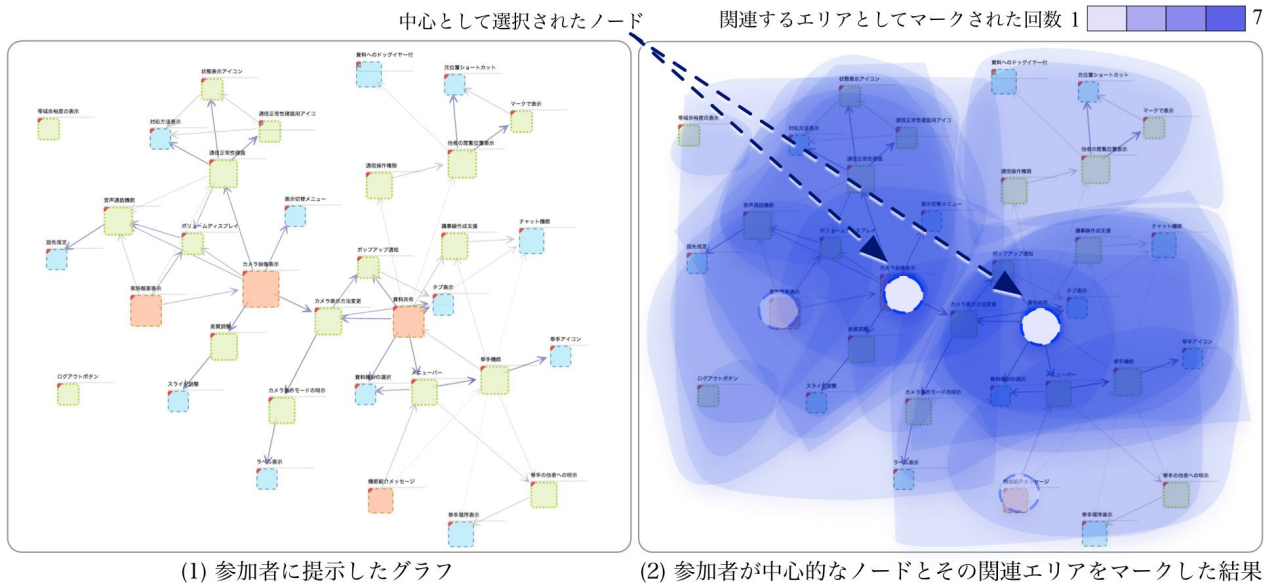


図 9 グラフ表現による可視化に関するテスト結果

Fig. 9 The result of marked visualized overview.

特に、中心的なノードとそのノードに関連するエリアを正しくグラフから読み取れるのかを検証し、さらに、読み取る際に、グラフのどの特徴を参考にするのかを分析した。調査には著者らの所属する組織の研究者7名（男性4名、女性3名）が参加した。

6.3.1 実施手順

まず、実験者があらかじめ UI-Filler を用いて、web ベースのビデオ会議システムに関するシナリオの作成とタグの抽出を行い、図 9(1) に示すようなグラフを得た。次に調査では、調査参加者にグラフのみを提示して、グラフにおいて中心的なノードはどれか、またそのノードに関連するエリアはどこかを、1分以内でマークするように調査参加者に依頼した。また、必要であれば中心的なノードは複数選んでもよいと教示した。さらに、作業終了後にインタビューを実施し、どのような理由で中心的なノードと、関連するエリアをマークしたのか質問した。

6.3.2 結果

図 9(2) に、調査参加者7名がマークした結果を重畳表示したものを示す。破線円は調査参加者がグラフの中心的なノードとして、その周囲の半透明青色は関連するエリアとして選択した結果である。色が濃いほど、多くの調査参加者が関連するエリアとしてマークした部分である。

図 9(2) に示したとおり、全員が2つのノードを中心として選択し、その2つのノードを中心として関連するエリアを描いた。ただし、1名の調査参加者は、追加で2つのノードを中心として選択した（薄い破線円）。また、2名の調査参加者は、選択した中心的なノードを含まないエリアを追加でマークした（図 9(2) 右上など）。

6.3.3 インタビュー

中心的なノードの判断について、第1に、ノードへの入

次数と出次数が多く、かつグラフにおけるノードの位置が中央に近いかが判断基準になったと全員が回答した。また、次数は関連するエリアの範囲を判断する材料にもなったと回答した。第2に、3名が橙色のノードに強く誘目されたと回答した。これは意図どおりの誘目を与えられていると考えられる。橙色のノードは出次数が多く、多くの他ノードに影響を与えるノードであるから、注意して分析することが求められるためである。第3に、1名が大きいノードは重要だと判断したとコメントした。シナリオの中で登場回数の多いタグを、ノードとしては大きく表現するように設計したので、意図した誘目を与えられたことが示唆された。

以上の結果から、UI-Filler が生成するグラフは、グラフの中心と関連するエリアとを素早く把握することを支援できると考えられる。つまり、シナリオのどこに中心的なタグ（要件）があり、それらに関連するタグ（要件）は何かを素早く把握するうえで、グラフは有効であると考えられる。

6.4 まとめ

3種類の調査結果を通じて、UI-Filler は3章で定義した3要点を支援することが示唆された。まず、シナリオと要件の管理については、調査1の結果から、シナリオの階層化とタグ付与の機能がシナリオ作成と要件管理を効率化できること、さらに調査2の結果から、タグがシナリオの分析を効率化することが示唆された。次に、グラフ表現を用いた可視化による複数シナリオの全体像の把握については、調査2においてグラフがGUI設計に活用される事例が観察され、さらに調査3の結果から、グラフの可視化結果の妥当性を示すことができた。最後に、シナリオとGUIとの対応関係の維持については、調査2の結果から、対応関

係維持の機能によって関連する情報への参照を早くし、シナリオを用いた UI の反復設計を効率化することが示せた。

7. 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効果的かつ効率的にできるツール、UI-Filler を提案した。具体的には、シナリオの階層化とタグの付与によって要件の明確化を支援する方法、およびタグどうしの関連を可視化することによって GUI 設計を支援する方法を新たに提案し、3 種類の調査を通じて有効性を考察した。

今後は、実際の GUI 改善事例に本ツールを導入しながら繰り返しツールを改善し、より本ツールの実用性を高めたい。そのために、すで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 を他のプロトタイプツールと連携できるようにファイルをエクスポートできるようにするなど、GUI 設計のさらなる効率化に寄与できるように取り組んでいきたい。

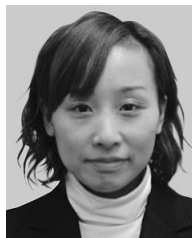
参考文献

- [1] ISO: ISO 9241-210 Ergonomics of Human-system Interaction – Part 210: Human-centred Design for Interactive Systems, Directly by ISO (2010).
- [2] Carroll, J.M.: *Making Use: Scenario-Based Design of Human-Computer Interactions*, MIT Press (2000).
- [3] Cooper, A., Reimann, R. and Cronin, D.: *About Face 3: The Essentials of Interaction Design*, John Wiley & Sons, Inc. (2007).
- [4] Hertzum, M.: Making Use of Scenarios: A Field Study of Conceptual Design, *International Journal of Human-Computer Studies*, Vol.58, No.2, pp.215–239 (online), DOI: 10.1016/S1071-5819(02)00138-6 (2003).
- [5] Nielsen, J.: *Usability Engineering, 1st edition*, Morgan Kaufmann (1993).
- [6] Nielsen, J.: Iterative User-interface Design, *Computer*, Vol.26, No.11, pp.32–41 (1993).
- [7] Cooper, A.: *The Inmates Are Running the Asylum*, Sams (1999).
- [8] Snyder, C.: *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*, Morgan Kaufmann (2003).
- [9] Gough, P., Fodermiski, F., Higgins, S. and Ray, S.: Scenarios-an Industrial Case Study and Hypermedia Enhancements, *Proc. 2nd IEEE International Symposium on Requirements Engineering*, pp.10–17 (1995).
- [10] Yanagida, K., Ueda, Y., Go, K., Takahashi, K., Hayakawa, S. and Yamazaki, K.: Structured Scenario-Based Design Method, *Human Centered Design*, Kurosu, M. (Ed.), Lecture Notes in Computer Science, Vol.5619, pp.374–380, Springer (2009).
- [11] 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).
- [12] Winkler, S. and von Pilgrim, J.: A Survey of Traceability in Requirements Engineering and Model-driven Development, *Software and Systems Modeling*, Vol.9, pp.529–565 (2010).
- [13] Sousa, K., Mendonca, H., Vanderdonck, J., Rogier, E. and Vandermeulen, J.: User Interface Derivation from Business Processes: A Model-driven Approach for Organizational Engineering, *Proc. SAC '08*, pp.553–560, ACM Press (online), DOI: 10.1145/1363686.1363821 (2008).
- [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] Rosson, M.B. and Carroll, J.M.: Integrating Task and Software Development for Object-oriented Applications, *Proc. CHI '95*, pp.377–384, ACM Press/Addison-Wesley Publishing Co. (online), DOI: 10.1145/223904.223953 (1995).
- [16] Obendorf, H. and Finck, M.: Scenario-based Usability Engineering Techniques in Agile Development Processes, *Proc. CHI EA '08*, pp.2159–2166, ACM Press (2008).
- [17] Tang, L., Yu, Z., Zhou, X., Wang, H. and Becker, C.: Supporting Rapid Design and Evaluation of Pervasive Applications: Challenges and Solutions, *Personal Ubiquitous Comput.*, Vol.15, No.3, pp.253–269 (2011).
- [18] Axure: Axure (2012).
- [19] Microsoft: Expression Blend (2012).
- [20] Li, Y. and Landay, J.A.: Activity-based Prototyping of Ubicomp Applications for Long-lived, Everyday Human Activities, *Proc. CHI '08*, pp.1303–1312, ACM Press (2008).
- [21] Sugiyama, K. and Misue, K.: Graph Drawing by the Magnetic Spring Model, *Journal of Visual Languages & Computing*, Vol.6, No.3, pp.217–231 (online), DOI: 10.1006/jvlc.1995.1013 (1995).
- [22] Kusano, K., Nakatani, M. and Ohno, T.: Scenario-based Interactive UI Design, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '13*, pp.391–394, ACM (online), available from <http://doi.acm.org/10.1145/2470654.2470710> (2013).



草野 孔希 (正会員)

デザインに関する研究に従事。ACM 会員。



中谷 桃子 (正会員)

2003 年早稲田大学大学院理工学研究科物理学及応用物理学修士課程修了。同年日本電信電話株式会社入社。現在まで、NTT サービスエボリューション研究所 (旧: NTT サイバーソリューション研究所) に所属。主にヒューマンコンピュータインタラクション, ユーザビリティ, ユーザ理モデルに関する研究に従事。ACM, ヒューマンインタフェース学会各会員。



大野 健彦 (正会員)

1994 年東京工業大学大学院理工学研究科情報科学専攻修士課程修了。同年日本電信電話株式会社入社。以来、視線インタフェース、視線測定技術、コミュニケーションの解明、遠隔地間コミュニケーション、熟達化支援、情報家電、ユーザエクスペリエンス、ユーザビリティ等の研究に従事。現在、NTT サービスエボリューション研究所、ACM、電子情報通信学会、日本認知科学会各会員。