

ウェブアプリ製品系列の UI プロトタイピング向け プロダクトライン開発

山崎 進^{1,a)} 山田 麻里衣^{1,b)} 高瀬 英希^{2,c)}

概要: ウェブアプリやウェブサイトなどでプロトタイピングを行う場合、ユーザーインタフェース (UI) やコンテンツに複数のバリエーションが存在し、それらを比較・評価したいことがしばしばある。そこで、このようなバリエーションをもつプロトタイプを一種の製品系列であると捉え、ソフトウェアプロダクトラインを適用する試みを行った。具体的には、ウェブサイトのバリエーションをフィーチャーモデルとして記述し、我々が静的なウェブサイトのプラットフォームとして採用する Ruby ベースの Middleman において可変性を実現する仕組みを開発し、実際にウェブサイトのデザインと実装を行った。

キーワード: ソフトウェアプロダクトラインエンジニアリング, UI デザイン, ウェブアプリ開発, フィーチャーモデリング

Software Product Line Engineering for UI Prototyping of a Web Application Family

YAMAZAKI SUSUMU^{1,a)} YAMADA MARI^{1,b)} TAKASE HIDEKI^{2,c)}

Abstract: Often, we develop prototypes of a web application or site, which has variation on user interface and contents, and then we want to compare and evaluate them. Thus, we regard such prototypes as a product line, and try to apply Software Product Line Engineering to them: we describe variation of them as a feature model, develop a framework that realize it using Middleman based on Ruby that we adopt as a platform of a static web site, and design and implement them using it.

Keywords: Software Product Line Engineering, UI Design, Web Application Development, Feature Modeling

1. はじめに

ウェブアプリやウェブサイトなどでプロトタイピングを行う場合、ユーザーインタフェース (UI) やコンテンツに複数のバリエーションが存在し、それらを比較・評価したいことがしばしばある。たとえば、我々の開発したイベントのウェブサイトの事例 [2] では、開発時点では UI のバリエーションを複数比較したいという要望があり、かつイベ

ントの進行とともにメニューやコンテンツに変化を設ける必要があった。

そこで、このようなバリエーションをもつプロトタイプを一種の製品系列であると捉え、ソフトウェアプロダクトライン [4][3] を適用する試みを行った。具体的には、ウェブサイトのバリエーションをフィーチャーモデル [3] として記述してみることを試みた。

本研究の目的は、ウェブアプリやウェブサイトのプロトタイピングで、UI やコンテンツのバリエーションを記述し、そのバリエーションの選択肢を与えてプレビューを表示したり、全てのバリエーションの組み合わせを一括でプレビューを表示したりするような静的ウェブページ構築の

¹ 北九州市立大学

² 京都大学

a) zacky@kitakyu-u.ac.jp

b) x6mcb015@eng.kitakyu-u.ac.jp

c) takase@i.kyoto-u.ac.jp

方法を提案することである。

アプローチは次のとおりである。

- (1) バリエーションを記述する方法としてフィーチャーモデル [3] を用いる。
- (2) 静的ウェブページの構築の手段としては Middleman[1] を用いる。Middleman を用いてバリエーションの選択肢を与えたり、全てのバリエーションの組み合わせを一括で表示できたりするようにする。

2. 研究の動機となった開発事例について

SWEST[2] では 2017 年から高瀬と山崎の手によりウェブサイトのリニューアルに着手した。高瀬が WordPress によるデザインと実装を行い、山崎が Ruby スクリプトによるプログラムページのデザインと実装を行って、これらを併用することによって公開に至った。プログラムページのデザインには山田も部分的に加わり、配色などを検討した。

イベント終了後に山崎が報告ページの開発を行った。Middleman[1] を採用することで、その高い記述性と再利用性により、たった数日程度の短期間で開発することができた。2018 年 2 月時点で [2] にて公開されているウェブページは、この段階のものである。

この経験を踏まえて、2018 年から Middleman を全面的に採用したウェブサイトのリニューアルを行うことを企画した。現在、開発中であり、2018 年 3 月のキックオフには公開する予定である。このデザインリニューアルでは、SWEST プログラム委員長の高瀬から、山田と山崎が要望のヒアリングを行い、山田が主にデザインを、山崎が主に Middleman による実装を行う。

デザインについては、単に意匠のデザインを行っただけでなく、情報デザイン(たとえば [5][6])を行った。すなわち、ヒアリング内容を元に、ウェブサイトで開示する情報を読み手にどのように伝えようと効果的かについて分析を行ってからデザインを行った。

3. デザインプロセスの進行とフィーチャーモデルの変化

本事例でフィーチャーモデルを導入した理由は、ヒアリングの結果、UI に関して実際にデザイン・実装して評価したいデザイン候補案が複数挙がったためである。ペーパープロトタイプでは良し悪しの判断がつかず、実際にデザイン・実装をして、ユーザビリティを評価してみないと、どの選択肢を採用したらいいかが判断できなかった。

また、複数の選択肢が要望として挙がってきたが、これらの選択肢をきちんと明文化する必要があり、かつ土曜な組み合わせが存在するのか、またそれらを網羅的に検討する必要性があった。

そこで、これを一種のプロダクトライン [4][3] だとみなし、可変性をフィーチャーモデル [3] で表現することで、明

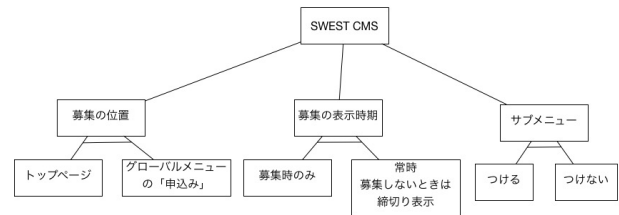


図 1 フィーチャーモデル (1)

Fig. 1 The Feature Model (1)

確に要望を定義しようとした。

最初にモデル化したフィーチャーモデルは図 1 である。たとえば、サブメニューについて、「つける」もしくは「つけない」のどちらかを選択するという、択一 (alternative) のフィーチャーが存在する、というような意味合いである。ここで、「募集の位置」「募集の表示時期」「サブメニュー」のような選択的なフィーチャーのことを可変点と呼ぶ。フィーチャーモデルは、どのようなバリエーションがあるか、すなわち可変性を明文化することができる。フィーチャーモデルを採用することで、可変性を明文化することができ、組み合わせを網羅的に検討することができる。

これを基に再度ヒアリングを進めたところ、さらに他の UI の可変性が明らかになり、さらに可変性としてはイベントの進行に伴って現れる可変性もあることが明らかになった。そのフィーチャーモデルは図 2 である。

当初このまま Middleman にて実装を行い、全ての可変性の候補の組み合わせをプレビューするウェブサイトの構築を行おうとした。一般に、可変点を削減することで、組み合わせの数を減らすことができる。

この段階でデザインの検討を進めたことで、可変点のうちの 1 つ「募集の位置」を削減することにした。この可変点は、参加募集・テーマ募集・インタラクティブセッション募集の 3 つ存在する募集ページについて、メニューを設けずにトップページに集約する方式と、グローバルメニューにそれぞれの募集メニューを表示する方式の 2 つの UI のバリエーションを比較検討するというものであった。デザインの検討を進めたことで、次のような理由でグローバルメニューにそれぞれの募集メニューを表示する方式の方が妥当であると判断したため、可変点にはせず固定にすることにした。

(1) それぞれの募集に関して、記述する内容が多い。現代においては、スマートフォンでの表示を最大限考慮してデザインする必要がある。もしトップページに各募集ページを羅列するようにデザインすると、スマートフォンで表示した時に長大なコンテンツを表示することとなり、扱いにくくなる。

(2) 他のページはグローバルメニューから個別のページへ遷移するようにデザインされている。もしトップページにそれぞれの募集についての記述を書いた場合に

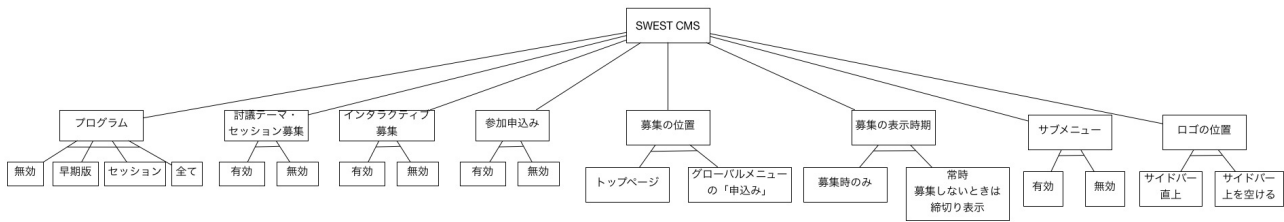


図 2 フィーチャーモデル (2)

Fig. 2 The Feature Model (2)

は、メニューから1クリックで遷移できるようにするためにトップページのサブメニューに各募集ページを掲載する必要がある。このようにすると、トップページのみサブメニューが存在することになり、他のページとルールが異なるので、ユーザーが混乱する懸念がある。

この点を踏まえて更新したフィーチャーモデルは図3である。これにより、可変点が7つ存在することが明らかになった。これらのうち「プログラム」「討議テーマ・セッション募集」「インタラクティブ募集」「参加申込み」がイベントの進行とともに現れるコンテンツのバリエーションを表し、「募集の表示時期」「サブメニュー」「ロゴの位置」がUIのバリエーションを表す。

「プログラム」の可変点については、バリエーションが4通りある。それ以外の可変点では、バリエーションがそれぞれ2通りある。全ての組み合わせは、256通り存在することが明らかになった。このうちUIのバリエーションについては、8通りの組み合わせがある。

4. 実装

デザインの実装には Middleman[1] を用いた。まず、フィーチャーモデルに現れる各フィーチャーをオプション変数として設定を与えることで、可変性をオプションに与えた通りに設定してプレビューと本番サイトを生成するスクリプトを開発した。

次に、考えられる全ての組み合わせを静的に生成し、URLの一部にオプションを記述することでバリエーションを選択するようなプレビューを生成するスクリプトを開発した。この全ての組み合わせを含むプレビューをプロトタイプとして顧客や想定ユーザーに提示することで、どのバリエーションが意図に近いあるいは利用しやすいかを評価する。

ただし、全ての組み合わせ (256通り) を生成すると時間がかかるため、今回はUIのバリエーション (8通り) のみの組み合わせを表示することに絞った。

今回のフィーチャーモデルでは、択一 (alternative) のフィーチャーしか現れなかったため、全ての組み合わせを網羅することが可能であった。もし任意 (optional) のフィーチャー、すなわち存在したりしなかったりを選択できるフィーチャーや、要求 (require) のフィーチャー、すな

わちあるフィーチャーを選択した場合、別のフィーチャーを必要とするような関係性が現れたとしても、組み合わせを網羅することは可能であろう。

ただし、容量などのパラメータの選択肢を与えるフィーチャーが含まれる場合には、代表的な選択肢をあげることで、択一のフィーチャーに変換してやる必要がある。

5. まとめと将来課題

UIやコンテンツに複数のバリエーションを持つウェブアプリやウェブサイトにて、フィーチャーモデル [3] を用いてバリエーションを記述することで、UIやコンテンツのバリエーションを明確にかつ簡潔に記述することができた。

次に、我々が静的なウェブサイトのプラットフォームとして採用する Ruby ベースの Middleman [1] において可変性を実現する仕組みを開発し、実際にウェブサイトのデザインと実装を行った。これにより、フィーチャーモデルで記述したバリエーションと同じ抽象レベルで、どの選択肢を選択するかオプションを指定することで、ウェブサイトの設定を簡潔に指定することができるようになった。また、プレビュー画面にて、全ての可能な選択肢を静的に生成することで、UIの比較検討を行いやすくすることができた。

将来課題としては、このような仕組みを再利用しやすくするために、Middleman のカスタム拡張 [1] を開発し公開することが挙げられる。

謝辞

SWEST 実行委員のみなさまには特に感謝します。また、ウェブサイトの開発にあたり多大な助言をくださった parcel チームの Devon Govett と Shawn Presser にも感謝します。

参考文献

[1] Middleman: Build static websites with an easy-to-use framework, Middleman (online), available from (<https://middlemanapp.com>) (accessed 2018-02-12).
[2] SWEST: 組込みシステム技術に関するサマワークショップ, SWEST 実行委員会 (オンライン), 入手先 (<https://swest.toppers.jp>) (参照 2018-02-12).
[3] Kang, K. C., Kim, S., Lee, J., Kim, K., Shin, E. and

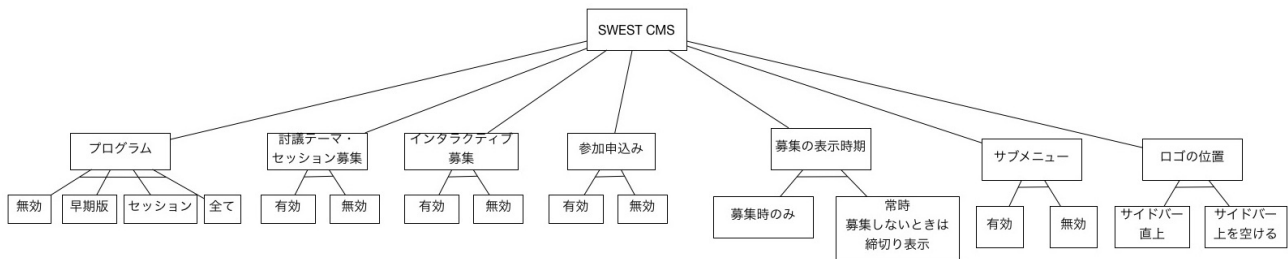


図 3 フィーチャーモデル (3)

Fig. 3 The Feature Model (3)

Huh, M.: FORM: A Feature-oriented Reuse Method with Domain-specific Reference Architectures, *Ann. Softw. Eng.*, Vol. 5, No. 1, pp. 143–168 (online), available from <http://dl.acm.org/citation.cfm?id=590631.590645> (1998).

- [4] Pohl, K., Böckle, G. and Linden, F. J. v. d.: *Software Product Line Engineering: Foundations, Principles and Techniques*, Springer-Verlag New York, Inc., Secaucus, NJ, USA (2005). 邦訳: 林 好一, 吉村 健太郎, 今関 剛 翻訳, ソフトウェアプロダクトラインエンジニアリング: ソフトウェア製品系列開発の基礎と概念から技法まで, エスアイピーアクセス, 2009.
- [5] 情報デザインフォーラム, 山崎和彦, 浅野 智, 上平崇仁 (編): 情報デザインの教室: 仕事を変える、社会を変える、これからのデザインアプローチと手法, 丸善 (2010).
- [6] 山崎和彦, 浅野 智, 安藤昌也, 上平崇仁, 木村博之, 小池星多, 原田 泰, 脇坂善則: 情報デザインのワークショップ, 丸善 (2014).