

エンドユーザによるユーザドキュメントの作成

Self generation of user documentation by end users

山本 喜一[†]

Yoshikazu Yamamoto[†]

[†] 慶應義塾大学 理工学部

[†] Faculty of Science and Technology, Keio Univ.

要旨

一般のアプリケーションプログラムはもちろんのこと、業務用の特注プログラムであってもそれらのユーザ用文書は、アプリケーションの開発者側が作成しているのが現状である。一方、開発者側はそのアプリケーションが使われる業務に必ずしも精通しているわけではなく、エンドユーザが本当に必要とするユーザ文書を適用するのは極めて困難である。本稿で述べるオンライン文書作成システムは、元々動的なオンラインヘルプを生成するための枠組みとして作成したものであるが、エンドユーザが自分たちのための文書を作成するためのツールとして利用することが可能であり、そこからアプリケーションの新しい利用方法を提案することもできる。

1. はじめに

我々の研究室では、1991年からいわゆるインテリジェントヘルプの研究を行っている。この研究は、ユーザの作業状況に応じて適切なアドバイスを行うことを目的としたもので、さまざまなアプリケーションに対していくつかのアプローチで研究を進めている。ユーザの状況に応じたヘルプを行うためには、従来のあらかじめ準備された固定的なヘルプでは目的を達成することはできないことから、ファクトベースとよぶ知識の断片から状況に応じたヘルプを動的に生成する枠組みを考案した。この枠組みを使うことで、いわゆる操作説明書に相当する文書を短期間に、しかも設計仕様が決まった段階から作成することができるようになり、従来行われていたアプリケーションのリリース直前の駆け込みによる文書作成作業を大きく改善できるようになった。

この枠組みは、複数のシステムから成っているが、その一つにテクニカルライターが使うことを意図して開発した、シナリオジェネレータがある。シナリオジェネレータは、テクニカルライターが実際にアプリケーションを使って目的とする作業を行い、そのときの画面の状態を保存しながら、作業履歴に基づいた作業シナリオを生成するものである。この類のシステムとしては、画像をキャプチャしながら操作手順を記録し、HTML、PDFなどの文書をシナリオとして生成するものや、操作手順そのものを動画として記録し、その上に指示や注釈を加えて、動画ベースの操作手順所を作成するものがある。これら既存のシステムと比較すると、我々の枠組みはヘルプの参照時にファクトベース利用して動的にシナリオに基づく操作手順を生成するので、シナリオ作成時の操作手順だけでなくもっと簡単な方法や、別の手順も生成できる点にある。例えば、ある機能を実行する方法として、メニューからの項目の選択とツールアイコンのクリックとが準備されていたとすると、従来の手法ではテクニカルライターが実際に行った方法がシナリオとして生成されるだけであるが、我々の枠組みでは両方の方法を併記することができる。

本稿では、我々のオンラインドキュメンテーションの枠組みを使ってエンドユーザが自分たちの利用法を説明する文書をどのように作成できるのか、またそれによってどんな効果が得られるのかについて考察する。

2. システムの構成

オンラインドキュメンテーションシステムの開発者から見た全体構成は、**図 1** に示すとおりである。この枠組みは、現在 ISO/IEC JTC1 SC7/WG2 において国際規格化を開始しようとしている。本来この枠組みが適用できるアプリケーションは、GUI に基づくアプリケーションであること以外に制限はなく、実際に OpenOffice Org. の表計算ソフトウェアである Calc、文書処理用の Writer、作図用の Draw、プレゼンテーション用の Impress のそれぞれのためのオンラインヘルプを作成した^{[1][2]}。さらに、仮想の Web

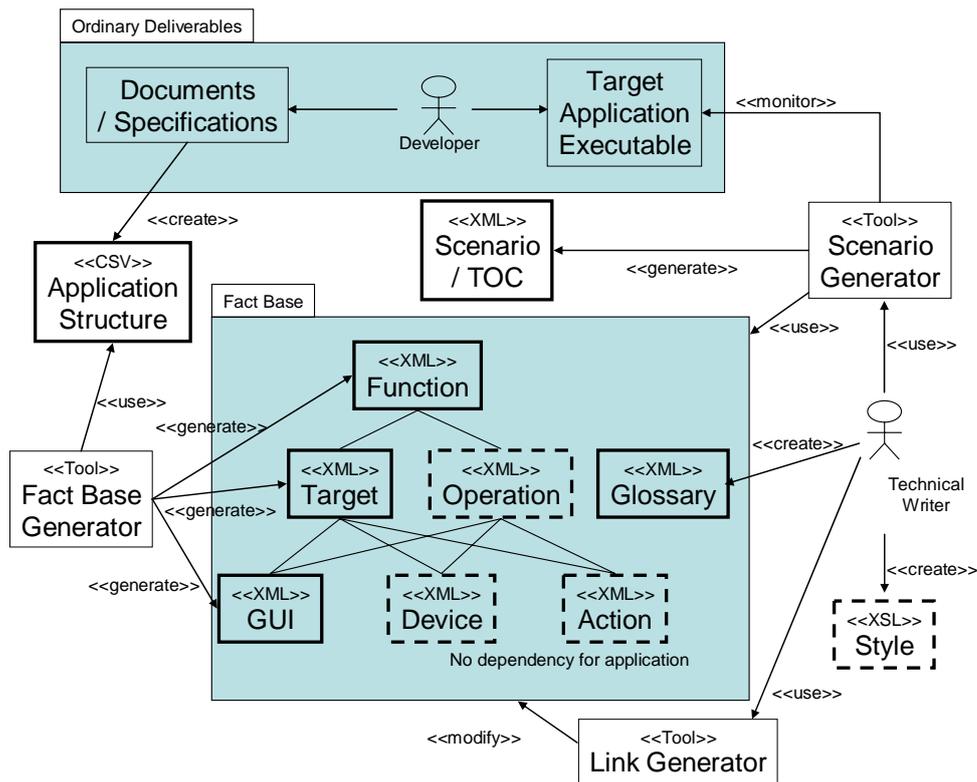


図 1 開発者から見たシステムの全体構成

アプリケーションとして出張旅費清算システムを作成し、そのオンラインヘルプも作成した。

図 1 に示す枠組みについて簡単に説明するが、特に下部の薄青く色付けした部分がファクトベースであり、それぞれの要素は XML のファイルである。ファクトベースについては次節で詳しく述べる。

開発者は、対象とするアプリケーションを開発するに当たりアプリケーションの機能と GUI についての仕様を確定するはずである。GUI 構成要素と機能との関係を表形式にまとめ、ファクトベース生成ツールによって XML 形式のファクトベースを生成する。テクニカルライターは、完成したアプリケーション又はプロトタイプ上でシナリオジェネレータを使って、ユーザが行う実際の作業を教授するためのシナリオを作成する。このシナリオも XML で記述され、必要に応じて説明文や画面をキャプチャした図を含めることができる。ここで作成したシナリオは、実際には使用した機能を順番に記録したものであり、ユーザがヘルプを要求したときにヘルプジェネレータによってユーザに提供する文書がそのたびに生成される。ユーザがヘルプを要求するたびに新たにヘルプ文書を生成することで、ユーザからの申告に基づいてユーザのレベルに応じた詳細さの粒度でヘルプを生成することができる。また、XSL で記述した表示スタイルを変更することで、通常の見出し形式と折り畳み形式とを簡単に切り替えることもできる。

さらに、あらかじめ準備しておいた用語辞書 (Glossary) を使ってヘルプ文書中に現れる用語の説明を表示するリンクジェネレータも使用可能である。

3. ファクトベース

図 1 の中心となっているファクトベースが、この枠組みの最も重要な部品である。ファクトベースは、Function, Target, Operation, GUI, Device, Action の 6 つの要素から構成され、それぞれ表 1 に示す内容をもつ XML 文書である。ファクトベースの記述様式は XML Schema で規定し、開発者が作成する CSV 形式のアプリケーション仕様から Fact Base Generator によって半自動的に生成できる。図 1 に示したファクトベース中で実線の枠で囲んだ要素はアプリケーションとの依存性が高く、アプリケーションごとにより直す必要がある部分である。一方、破線の枠で囲んだ要素はキーボードとマウスを使う普通の使用

表 1 ファクトベースの構成要素

Function	アプリケーションの機能
Target	アプリケーションの操作対象となる、表のセル範囲、文字、段落などのオブジェクト
Operation	ユーザが行う可能性のある、クリック、選択などの操作
GUI	アプリケーションのウィンドウの構造とその構成要素の静的な表現
Device	アプリケーション操作のためのキーボード、マウスなどの入力機器
Action	Device で実行する、クリック、キー入力などの基本操作

環境ではほとんど同じなので、一度作成すれば再利用できる^[3]。また、用語辞書である **Glossary** を作成しておくと、**Link Generator** によってファクトベース中の用語からハイパーリンクを付けることもできる。

アプリケーションのメニュー構造は、ほとんどの場合階層的になっていることから、**CSV** 形式（表）で簡単に表現できるが、同じ機能がツールアイコンなど別の **GUI** 構成要素で提供されている場合など、表形式ではうまく表せないこともある。そのため、基本的には表形式で作成したアプリケーション仕様を使うが、各機能個別に詳細な属性を指定するためのツールも準備している。

OpenOffice.org のファクトベースを作成した結果、**Writer**、**Calc**、**Impress**、**Draw** の各アプリケーションに対して、アプリケーション仕様は当然異なるものの、**Operation**、**Device**、**Action** はほとんど変更なしに共用でき、アプリケーション仕様も最初に作成した **Calc** を参考に簡単に作成できた。

4. シナリオの作成とヘルプの表示

テクニカルライターは、アプリケーション作成者から与えられた情報や、完成したアプリケーションを使ってユーザ文書のためのシナリオを作成する。シナリオは、アプリケーションが提供する機能を使って業務を行う際の手順書に相当するもので、シナリオ作成のためのソフトウェアもフリーのものを含めていくつかは実用化されている^{[4]~[9]}。これらのソフトウェアは、アプリケーションの操作手順をユーザに分かりやすく示すことを目的とし、自動的な画面キャプチャを主体とした静止画に基づくシナリオと、動画を主体とするシナリオに大別される。当然のことながら、動画を主体とするシナリオは操作手順を逐一フォローするため、ヘルプを参照することで操作の詳細を知ることができる反面、そのための時間が掛かり、ヘルプのサイズも大きくなってしまう。一方、静止画を主体とするシナリオでは、ユーザがヘルプを読み、説明されている操作を理解しなければならない。

これらのシナリオ生成ソフトウェアと我々の枠組みとの本質的な相違は、我々の枠組みがファクトベースを用いてヘルプの参照時に動的にヘルプ文書を生成するのに対し、従来の方法はあらかじめ作られているヘルプ文書を表示するだけであることである。ここでは詳しくは述べないが、我々の枠組みを使えばユーザのレベルに応じたヘルプ文書を生成することができ、実際に今までの実現例ではユーザからの要求に応じて 4 レベルの詳細さでヘルプ文書を生成している。ユーザの知識レベルを正確に認識することができれば、ヘルプシステムがユーザの技量に応じた的確な詳細度のヘルプを表示することが可能となる。

Calc や **Writer** のような複雑なアプリケーションでは、ひとつの機能を複数の方法で実行できるようになっている。例えば、文字列の複製と貼り付けはメニュー、ツールアイコン、ショートカットキーのいずれでも機能を起動することができ、それらの組み合わせで処理を進めることができる。すなわち、複数の機能を一連の操作によって実行して行う作業に対して、実際の操作列をすべて列挙するのは大変な作業になってしまう。したがって、操作手順を前もって記録する従来のシナリオ作成では、テクニカルライターがそのときに行った操作手順だけが表示されることになる。

我々の枠組みにおけるシナリオ作成は、**エラー! 参照元が見つかりません。** に示す流れで行う。ここでは、**Writer** 用の **Scenario Generator** を示しているが、どんなアプリケーションに対しても同じ構造であ

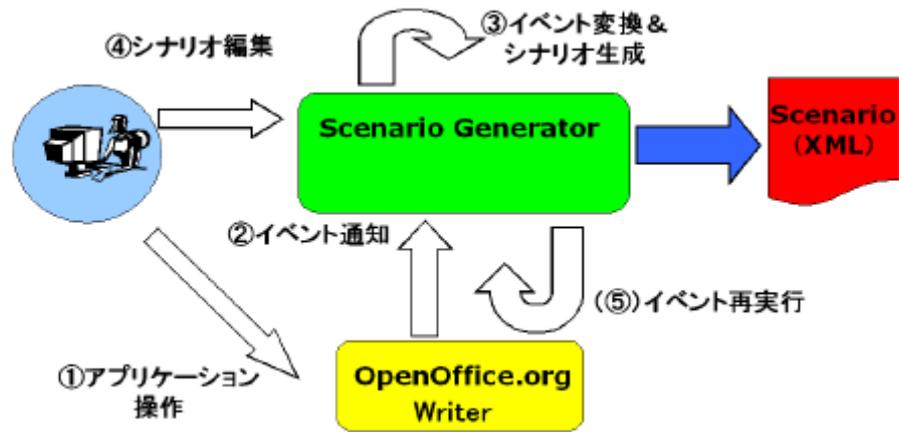


図 2 シナリオジェネレータ

る。Scenario Generator はアプリケーションを操作したときのイベントを引き金として対応する機能のシナリオを書き出し、必要ならば画面のキャプチャも行う。一度作成したシナリオを編集する機能ももっている。

一般にシナリオの作成は、テクニカルライタの仕事であるが、この枠組みを利用したときにはユーザー側のしかるべき人が作業手順書や操作手順書を作成することもできる。Scenario generator がどれだけ使いやすいとされているのかに依存するが、いわゆるパワーユーザや業務専門家などユーザー側の担当者がエンドユーザ向けの文書を作った方がよい場合がある。特に、ある業務用に開発するアプリケーションでは、ユーザー側からの要求しように基づいて機能仕様を決定し、アプリケーションを開発するのが普通である。このとき、業務に依存した機能仕様を完全に伝えることが難しく、ユーザー側でさまざまな工夫を行って業務をこなすことも多い。このような場合には、その操作説明書を開発者側で作成することは難しく、ユーザー側でエンドユーザ向けの文書を追加しているケースが多い。

現在開発中の Scenario generator は、生成する XML ファイルや画像ファイルの格納場所を前もって指定し、ユーザが簡単に管理できるようにすることで、ユーザー側でのシナリオ作成が簡単にできるように工夫している。業務用のアプリケーションに関しては、業務の専門家がその使い方を最もよく知っていることが多く、ユーザー側でシナリオを作れるようにすることで、エンドユーザの教育などを効率よくすることが期待できる。

5. まとめ

オンラインヘルプ生成システムの枠組みを利用して、ユーザー用文書をユーザー側で作成する可能性とその効果をさらに考察する必要がある。我々は、ユーザー用文書を簡単にかつ動的に生成する枠組みを提案、実現したが、この枠組みをどのように利用するかについてはさらに研究を進める必要がある。

参考文献

- [1] 品川一貴, 平井航一, 山本喜一: ユーザ文書のオンライン生成, 情報システム学会第 1 回研究発表大会論文集(2005).
- [2] 平井航一, 辻将悟, 山本喜一: ユーザグループによるユーザ文書の生成, 情報システム学会第 2 回研究発表大会論文集 (2006).
- [3] 辻将悟, 山本喜一, 平井航一: ファクトベースを用いたヘルプドキュメントの生成, 情報処理学会論文誌, Vol. 48, No. 8, pp. 2699-2712(2007-08).
- [4] SoftSimulator, <http://www.softsimulator.jp/>
- [5] Adobe Captivate 3, <http://www.adobe.com/jp/products/captivate/>
- [6] CamStudio, <http://nonn-et-twk.dnsalias.net/twk/CamStudio/>
- [7] Wink, <http://www.debugmode.com/wink/>
- [8] Click! レコーダー, <http://www.gluesoft.co.jp/ClickRec/>

[9] ASTRAEA, <http://www.vector.co.jp/soft/winnt/business/se432118.html>