

ユーザグループによるユーザ文書の生成

Creation of user documentation by a user group

平井 航一[†] 辻 将悟[‡] 山本 喜一[‡]
Koichi Hirai[†] Shogo Tsuji[‡] Yoshikazu Yamamoto[‡]

[†]慶應義塾大学大学院 理工学研究科

[†] Graduate school of Science and Technology, Keio University

要旨

チュータ、ウィザード、ヘルプなどのアプリケーションのユーザ文書を、ユーザからの要求に応じてオンラインドキュメントとして動的に生成、表示するためのフレームワークと、OpenOffice.org の Calc を例に実現したシステムを昨年度の大会で紹介した。本年度は、Calc の外、Writer, Draw, Impress にこの方式を適用するよう拡張を図っている。さらに、本来テクニカルライター用に設計したシナリオジェネレータを、さらに使いやすくすることによって一般ユーザが利用者支援の文書のシナリオを作成できるように工夫し、ソーシャルネットワークと同様の機構を利用して、ユーザ文書の効率的な作成を目指す。

1. はじめに

複雑な機能を持つアプリケーションが数多く使われるようになり、その一方でコンピュータのユーザは小学生から高齢者まで広がって、ユーザのスキルレベルがかつてないほど広がっている。多くのアプリケーションでは、従来の分厚い紙のユーザマニュアルに代わり、いわゆるオンラインドキュメントが多く使われるようになってきた。残念ながら現状では、紙の文書をそのままオンラインファイルとして利用したり、精々ハイパーテキストの機能を持たせたりしているに過ぎないものも多い。

われわれが昨年提案したオンライン文書作成システム^[1]は、現在のほとんどのアプリケーションで使われている GUI ベースの作業に基づき、ユーザがアプリケーションを利用するときに、操作対象を指定し、機能を実行するための操作を行うというオブジェクト指向モデルを利用し、ヘルプ、チュータ、ウィザード、参照用文書などすべてのユーザ文書をオンライン文書として自動的に生成する。また、ヘルプ、チュータ、ウィザードについては、ユーザが自分のスキルレベルを選択することによって、ユーザに合わせた粒度のオンライン文書を動的に生成するものであった。

ユーザにとっては、現在の固定的な文書に比べ自分のスキルレベルに合わせて動的に生成された文書を得られるという利点があるとともに、アプリケーション作成者、特にユーザ文書を作成しているテクニカルライターにとっては、同様な記述を何度も繰り返さずにすむだけでなく、アプリケーションリリース直前の集約的な作業を軽減し、アプリケーションの修正変更に伴う文書の変更にも柔軟に対応できることが大きな特徴となっている。

本研究は、現実のアプリケーションに応用することを目指し、昨年度は OpenOffice.org の表計算アプリケーションである Calc を対象として実現したが、今年度はさらに Writer, Draw, Impress のいわゆるオフィスツールを対象を広げている。昨年度までは、アプリケーションの操作方法に関する文書の作成だけを目的としていたが、用語辞書に対応するファクトベースを新たに作成し、ユーザ文書から項目をリンクすることによって、概念の説明にもある程度対応できるようにしている。また、テクニカルライターの使用を前提として開発したシナリオジェネレータを改良し、一般ユーザが発見した利用方法をシナリオとして提供してもらうことによって、ユーザ文書をさらに充実するとともに、ソーシャルネットワークと同様の機構を使うことによって、善意のユーザからのフィードバックを積極的に文書作成に取り込むことの効果を評価する予定である。

2. ファクトベース

ユーザに提示されるヘルプ、チュータ、ウィザードのそれぞれの項目をユーザの行う作業とし、その作業を説明する記述をシナリオと呼ぶ。シナリオは、テクニカルライターが記述するものであるが、基本的にはメニューやアイコンを用いて対象となるオブジェクトを操作することの繰り返しからなる操作列に

よって表される。シナリオの作成については、5.で述べる。

シナリオを表現するために、我々は図 2 機能のファクトの要素に示すファクトベースと呼ぶ構造を定義した。ファクトベースは、最も細かな粒度をもつ対象と動作を末端のノードとする階層構造として、アプリケーション作成者が定義するものとする。図 2 機能のファクトの要素のファクトベースの構成要素を次のとおりに定義した。

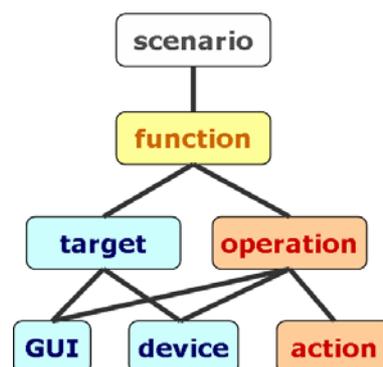


図 1 ファクトベース

- (1) GUI 構成要素 (GUI) : ウィンドウ, ダイアログボックス, メニュー, ボタンなどアプリケーションの画面を構成する要素。階層構造を持っているのでアプリケーションの機能が決めれば内容を決定できる。これを表形式で表現することにより XML 記述を自動生成するツールを作成した。
- (2) デバイス情報 (device) : 106・101 キーキーボード, 2 ボタン・3 ボタンマウスなど入力機器及びそれらを構成するキー, 左ボタン, 右ボタンなど。人手によって作成する必要があるが, 新たなデバイスを追加しない限り再利用できる。
- (3) 基本動作 (action) : マウスボタンを“押す”, “放す”, “クリックする”, あるキーを“押す”などユーザが行うそれ以上分解できない動作。ダブルクリックは“クリックする”を2回連続する動作なので, 基本動作としては定義しない。入出力のデバイスに依存するので, デバイス情報と同様に人手によって作成する必要があるが, 新たなデバイスを追加しない限り再利用できる。
- (4) 対象 (target) : GUI 構成要素やデバイス情報では表現しきれない操作対象を定義する。たとえば, “複数のセル”や“セル範囲”は, GUI 構成要素のひとつであるセルの集合として定義する。Writer では, “文字列”や“段落”が対象となるので, アプリケーションによって異なる定義が必要となる。
- (5) 操作 (operation) : 基本動作では表現できない動作で, 複数の動作の組合せ, デバイス情報と基本動作の組, GUI 構成要素と基本動作の組となる。たとえば, “ドラッグする”という操作は, “マウスの左ボタンを押す”, “マウスを移動する”, “マウスの左ボタンを離す”によって表現できる。これも一度定義しておけば変更がないので再利用できる。
- (6) アプリケーション機能 (function) : アプリケーションが提供する機能。メニューやツールアイコンによって表現されている機能に対応する。GUI 構成要素を表形式で表現することによってやはり自動的に XML 記述を生成するツールを作成した。

シナリオを含むファクトベースはすべて XML によって記述している。また, それぞれのファクトの XML 記述を形式的に定めるために, XML スキーマ記述言語である XML Schema を用いた。たとえば, アプリケーション機能のファクトに対する XML Schema は, 図 2 のように模式化できる。それぞれのアプリケーション機能に対して機能 (function) 要素を定義し, それぞれの機能を id 属性によって区別し, 内容を記述する要素として, 多言語対応のための言語要素(lang)を置き, その下に名前(name), 説明(explanation), イメージ(image), 動画(movie)の各要素を持たせている。

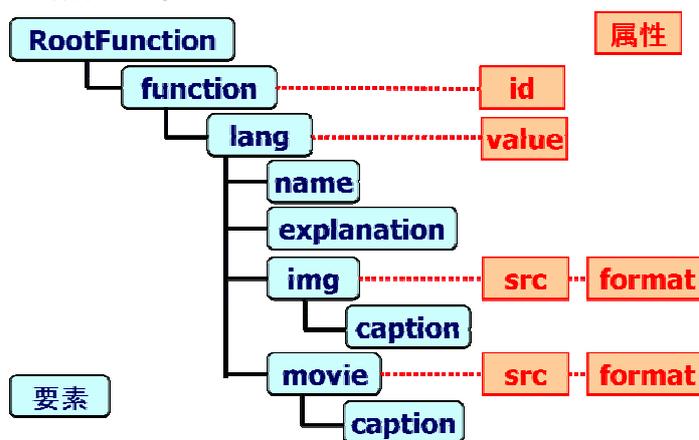


図 2 機能のファクトの要素

図 2 の XML Schema に従った“セル範囲の選択”機能の XML 記述は, 図 3 のとおりである。図 3 に示すとおり, セル範囲を選択するとき, マウスをドラッグして選択する方法と, Shift キーを押しながら

らマウスの左ボタンをクリックする方法とがあるが、それらを<case>タグによって表現している。図 3 には言語要素が含まれていないが、言語要素は多言語対応のために含めた要素で、アプリケーション機能の記述自体には影響しないため、言語要素だけを別に記述することで機能記述の簡潔化を図っている。

3. オンラインヘルプの動的生成

我々のシステムの現在の実現では、図 4 に示すとおり Apache Tomcat をサーバとして利用し、JSP (Java Server Pages)を使ってユーザからの要求によってヘルプを動的に生成している。この方式は、一つの実現例として示したものであり、これが唯一の実現方法ではない。昨年提案したように、図 2 機能のファクトの要素に示したファクトベースの構造を用いることによって、実現方法によらずレベル別に表示を変えることや多言語のドキュメントといった柔軟なオンラインドキュメンテーション環境が構築できる。

4. 用語辞書とリンク

本システムでは、昨年まではアプリケーションの操作方法に関する文書の作成だけを目的としていたが、オンラインヘルプから提供される情報として、アプリケーションの操作方法だけでは不十分である。そこで新たに用語辞書に対応するファクトベースを作成している。この用語辞書の項目へオンラインヘルプの項目からリンクすることで、用語や概念の説明にある程度対応できるようになる。

オンラインヘルプと用語辞書をリンクさせる際、テクニカルライターが手動でリンクを作成しなくてはならないため手間がかかってしまう。そこでヘルプの記述内容を形態素解析し、そこで使用している用語と用語辞書の中の項目を対比することにより、リンクの負担を軽減する仕組みを作成している。リンク付けを完全に自動化することはほとんど不可能であるが、この仕組みでは一度定義したリンクを保存することにより、新たなシナリオやファクトが出来た際にも、その部分だけのリンクを作成すればすむようになる。

5. シナリオの作成

本システムでは、ユーザに実際に表示されるドキュメントは、テクニカルライターが記述するシナリオに基づいて動的に作成することになる。テクニカルライターのシナリオ作成が簡単でなければ、結局は従来のドキュメント作成と同じことになってしまう。そこで我々は実際にアプリケーションを使用して、シナリオに従った操作を行い、そのときのイベントを獲得して自動的にシナリオの骨組みを生成するシナリオジェネレータを作成

```
<?xml version="1.0" encoding="Shift_JIS"?>
<RootFunction>
  <function id="SelectCellRange">
    <case num="1">
      <step num="1">
        <componentid="RootWindow_CellField_Cell"/>
        <operation id="LeftClick" />
      </step>
      <step num="2">
        <device id="Mouse" />
        <operation id="Drag" />
      </step>
    </case>
    <case num="2">
      <step num="1">
        <component id="RootWindow_CellField_Cell" />
        <operation id="LeftClick" />
      </step>
      <step num="2">
        <device id="ShiftKey" />
        <action id="Pressing" />
      </step>
      <step num="3">
        <component id="RootWindow_CellField_Cell" />
        <operation id="LeftClick" />
      </step>
    </case>
  </function>
</RootFunction>
```

図 3 セル範囲選択機能の XML 記述

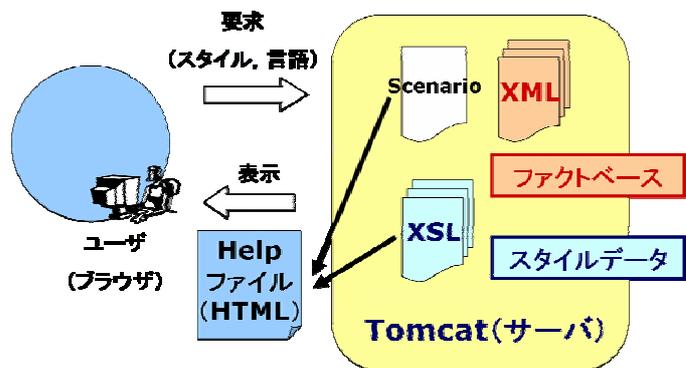


図 4 オンラインヘルプの自動生成

した。

シナリオジェネレータで実際に作成したシナリオを図 5 に示す。アプリケーションで利用した機能を知り記録するために OpenOffice.org のマクロ機能を使用しているので、作成したシナリオは再生することも可能である。この機能により、多言語対応ヘルプを生成する場合でも一度作成したシナリオを他言語のアプリケーションで再生することによりその言語用のシナリオを作ることができる。アプリケーションのバージョンが更新されても、機能名が変わっていなければシナリオの再生は可能である。またシナリオ作成時に画像をキャプチャしたタイミングを記録しているので、シナリオ再生時には同じタイミングでキャプチャが行われる。この機能により、アプリケーションのバージョン更新時にシナリオを再生するだけで新しいバージョンの画像を取得できる。他言語版でも同様のことが可能であるが、言語に依存する部分は人が書き換えなければならない。

シナリオには機能列だけでなく、説明の文章が必要となる場合がある。シナリオジェネレータは、必要なときには人間が説明のための sentence を挿入できるようになっている。

6. ユーザによるドキュメント生成

シナリオジェネレータをユーザに使用できるようにすることで、ユーザは自らの手でヘルプのシナリオを作成することができるようになる。ヘルプシステムを実装したサーバに、作成したシナリオをインポートすることによって、新しいヘルプトピックを追加できる。このためのサーバ側の仕組みとシナリオジェネレータの改良を現在行っている。このシステムにより、テクニカルライターが作成したヘルプに不足があった場合や、ユーザが発見したアプリケーションの新しい使い方をオンラインヘルプに加えることが可能となり、ユーザ文書をさらに充実させることが可能となる。

7. まとめ

ここで提案した枠組みは、特定のアプリケーションを対象としたものではなく、さらに特定の實現方法に限られるものでもない。図 2 機能のファクトの要素のファクトベースの構造を用いることによって、ユーザドキュメントを動的に生成でき、ユーザ、アプリケーション開発者双方にとって利点の大きいことを示した。さらに、シナリオジェネレータを使用することにより、テクニカルライターだけでなくユーザも簡単にシナリオを作成し、ユーザ自らユーザ支援文書の作成に参加できる。これまでのテクニカルライター中心のドキュメンテーションでは作られなかったユーザ指向のユーザ支援文書が作られるようになる可能性がある。

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, A-09-1-4, 2005 年 11 月.

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<RootScenario lang="ja">
  <scenario id="WritingMulti-lineText">
    <title>テキストを複数行にする</title>
    <sentence>セルの右端でテキストを自動的に折り返すには、次の操作を行います。</sentence>
    <step>
      <function id="GoToCell" />
      
      <function id="WrapText" />
      
      <sentence>Ctrl キーを押した状態で Enter キーを押すと、任意改行ができます。この操作は、入力バーにではなく、直接セルに入力している場合にだけ有効です。</sentence>
    </step>
    <topics>
      <related_topic id="WritingMulti-lineText" />
    </topics>
  </scenario>
</RootScenario>
```

図 5 作成したシナリオの XML 記述