Web を利用したソースコードと解説の効果的な共有手法

An effective way to share source code and explanations using the Web

寒河江惣貴 †, 福田浩章 †
Soki Sagae†, and Hiroaki Fukuda†
† 芝浦工業大学 工学部
†Faculty of Engineering, Shibaura Institute of Technology.

要旨

近年，インターネットは様々な分野で大きな役割を担っており，プログラミングに関しても同様である．しかし，現在ソースコードやその解説が掲載されている主要なサイトでは，ソースコードと解説は併記できず，読むためにスクロールを要する．先行研究である CodeWiki は，解説を画面内に別のウィンドウを用いて表示することでスクロールを排することができると提案した．しかし，Wiki システムに則った CodeWiki は，本文以外も表示されてしまうために他ページからの参照に適しておらず，既存のサービスからの利用は難しい．

そこで本研究では，スクロールすることなくソースコードと解説を同時に表示でき，かつ他ページからの参照に適した Web サービスを提案する．

1. はじめに

近年，様々な分野でインターネットは大きな役割を担っている．プログラミングに関しても，多くのサイトにソースコードやその解説が掲載されている．現在主にソースコードや解説の掲載に使用されている note[1]や Qiita[2]では，解説はソースコードの上下に併記している [1]．これは Qiita 等が採用している Markdown[3] が左右併記に対応していないためであり，閲覧者はソースコードと解説を同時に読むことができず，スクロールすることを余儀なくされている．

この問題を解決するために提案されたのが CodeWiki[1] である．CodeWiki では，既存の Wiki の文法にタグを追加することによってソースコードと解説を併記できるようにしている．しかし，Wiki は iframe 要素を利用して埋め込むと，メニューなどの本文以外の余計な情報も参照されるため，他ページへの埋め込みに適さない．既存のサイトに掲載された記事は，ソースコードや解説だけでなく，動機やコメントなどの要素を含めて成立している．CodeWiki では既存の記事の全ての情報をソースコードと解説と同じウィンドウで読むことはできないため，記事の閲覧者は記事を読み進めるために新しいウィンドウやタブを開かなければならない．

そこで本研究では，ソースコードと解説を左右に併記でき，かつ埋め込みを利用した参照に適した新たな Web サービスを提案する．

2. 先行研究

Wiki は「HTML タグよりも簡単な特殊タグが用意されているため，編集者は自分の考えを素早く表現することができる」[2] という考えのもと，CodeWiki ではソースコードとその解説を並行して表示できるように，既存の Wiki に新しいタグのセットと，解説とソースコードを記述するための編集画面を追加して提案された．CodeWiki で追加されたタグによって，閲覧者は画面をスクロールすることなく，ソースコードの横に解説を表示して読むことができる．このとき常にすべての解説を表示していると，情報量の多さからむしろ閲覧者の理解を妨げる．そのため最初は解説を表示せず，閲覧者がソースコードをクリックしたときにそのコードに対応する解説を表示する．また，対応するソースコードと表示された解説の外枠を同じ色にすることによって，複数の解説が表示されていてもどの部分の解説なのかが一目で分かるよう配慮されている (図 1)．

また，CodeWiki はソースコードと解説の記述者に対し，編集時にソースコードと解説をそれぞれ個別のウィンドウに記述させるために，ウィンドウを二つ備えた独自の編集画面も提案している．

[1] https://note.com/
[2] https://qiita.com/

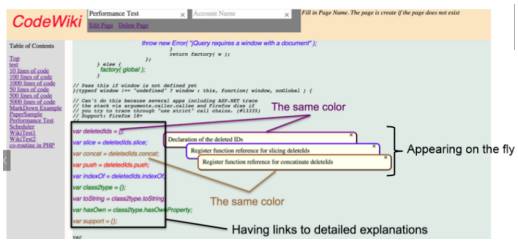

図 1: CodeWiki の閲覧イメージ

このように CodeWiki は，ソースコードと解説を画面スクロールすることなく読むことができないという問題を解決している．しかし，Wiki の形式をとったことによって，メニューバーなどの本文以外の情報が存在するため，他ページからの参照の際に本文以外の情報も表示されてしまい，既存のページに埋め込んで使用する方法への適性が低い．現在ソースコードと解説が掲載されている記事の大部分は，ソースコードや解説以外にも，リンクや参考文献，コメントなどを含めて一つの記事として成立しているが，CodeWiki を使用した場合，それらの情報をソースコードや解説と同時に読み進めるためには，ウィンドウやタブを複数開かなければならない．これは記事の閲覧者にとっては手間となる．

3. アプローチ

本研究では，CodeWiki で提案された仕組みの利点を引き継ぎつつ，既存のサービスからの参照に適した Web サービスを提供する．CodeWiki はソースコードと解説を掲載するためのページと，そのページを記述するための編集画面の二つを提案している．それに倣い，本研究でもソースコードと解説を読むための閲覧用ページと，それらを記述するための記述用ページを用意する．

3.1. 閲覧用ページ

CodeWiki と同様にソースコードと解説を同時に読むことを可能にするため，閲覧者が解説文が書かれているウィンドウの位置や大きさを変更でき，かつ同時に複数開くことも可能にする．閲覧者はソースコード中に表示されるボタン (解説表示ボタン) をクリックすることによって，対応する解説を表示することができる．

記述するソースコードの量が多い場合は，必然的に解説の量も多くなる．それに応じて閲覧者は，解説を複数開いた場合に，どの解説がソースコードのどの部分に対応するのか見失いやすくなる．したがって解説が表示されている間は，表示されている解説が指す範囲のみ枠で囲む (図 2)．このとき，ソースコードを囲む枠線の色と，対応する解説のウィンドウの上部の色は同色を用いる．これによって閲覧者には，ソースコードのシンタックスハイライトを崩さず快適な可読性を保持したままに，どの解説がソースコードのどの部分にあたるのかを視覚的に示すことができる．そのため，解説を多数表示した状態でも，対応したソースコードと解説の組み合わせが見失いにくくなる．また，閲覧用ページは既存のサービスに iframe 要素を使用して埋め込むことができるようにする．このとき，ソースコードと解説の可読性を維持するため，ソースコードと解説以外の情報は含めない．

図 2: ソースコードと解説の表示イメージ (プロトタイプ)

3.2. 記述用ページ

記述用ページは，ソースコードと解説の記述者に負担を与えずに，上記の閲覧用ページへと変換できる形式で記述できるよう，ソースコードと解説を記述するページである．記述用ページでは CodeWiki

と同様に，記述者の混乱を避けるためにソースコード用と解説用の 2 つのテキストエリアを用意する (図 3)．記述者はソースコードと解説を，それぞれのテキストエリアに分けて記述する．記述用ページでは，Undo と Redo に加えて，記述者を補助する機能を呼び出せるボタン群をページ上部に配置する．ボタンで呼び出せる補助機能は大別して以下の 2 種類がある．

1. タグ挿入
   閲覧者は解説表示ボタンをクリックすると解説を表示することができるが，記述者がこのボタンを直接 HTML で記述する方式では，記述者の手間と負担を増加させる．これを回避するため，本研究では記述者がソースコードで解説する部分を [/desc (個別の解説タイトル)](ソースコードの該当部分)[/desc (個別の解説タイトル)] というタグ (解説挿入タグ) を用いて記述することで，Web サービスに保存する際に自動でボタンに変換される．解説の内容とタイトルは関連づけられ，同一のタイトルを使用すれば複数箇所に同じ解説を挿入できる．

   記述者は自身でタグを全て記述しなくても，ページ上部のタグ挿入ボタンをクリックすればソースコードと解説双方のテキストエリアにタグを挿入できる．タグを挿入するとき，テキストボックスにあらかじめ挿入したい解説のタイトルを入力しておくことで，タグは解説タイトルが反映された状態で挿入される．

2. プレビュー機能
   記述者への具体的な出力イメージのフィードバック，および記述ミスの削減のため，記述用ページには出力時のプレビュー機能も用意する．記述者がプレビュー表示用ボタンを押すことで，現在の記述で出力した場合のイメージが表示される．記述者はプレビューを記述しているページの下半分，もしくは別のタブのどちらに表示するか選択できる．

記述者は記述したソースコードと解説を，閲覧者用のページとして Web サービスのサーバに保存できる．そして，保存したページは埋め込み参照が許された既存のページに埋め込むことができる．

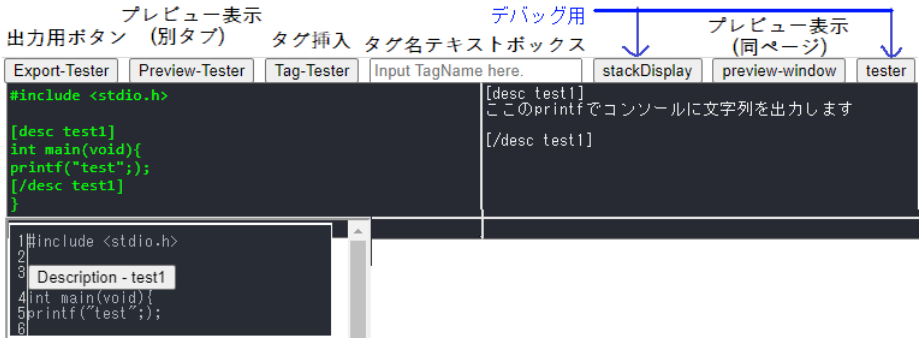

図 3: 記述用ページのイメージ (開発中のプロトタイプ)

4. 実装

本研究の Web サービスは HTML によるページ構成と，JavaScript を用いた DOM 操作によって実現する．

4.1. 閲覧用ページ

閲覧用ページは，記述者が書いたソースコード本文と解説の文字列を Web サーバに保存するときに HTML 要素内に保持し，それらの情報を元に DOM 操作で整形することによって作成される．閲覧用ページのロード時に，ソースコード文字列を HTML 要素から読み込み，文字列置換処理を用いて解説挿入タグを解説表示ボタンに変換し，画面に表示する．ソースコードには highlight.js[3]を適用し，シンタックスをハイライトする．同時に，HTML 要素内から解説文字列を読み込み，囲まれたタグごとに分割す

[3]https://highlightjs.org/

る．分割した解説には開始行番号と行数の情報を付け加え，解説タイトルと関連づけてメモリに格納する．解説表示ボタンがクリックされたら，解説タイトルと関連づけられた解説をメモリから読み込む．

解説の表示には JSFrame.js[4]を利用し，ユーザが自由にその大きさや位置を変更でき，複数開くこともできるフローティングウィンドウを用いて表示する．解説の表示時には，Web サービスであらかじめ用意した数種類のうちから色を選び，解説上部の色を決定する．解説はそれぞれ開始行番号と行数を保存しているため，これを利用して，ソースコードのうち解説が示している部分のみを，対応したフローティングウィンドウと同色の枠で囲む．

4.2. 記述用ページ

記述用ページは，ソースコードと解説の二つのテキストエリアと，記述補助用のボタン群で構成する (図 3)．記述用ページはあらかじめ HTML とスタイルシートによる記述で整形してあり，ブラウザのウィンドウサイズに合わせて大きさを変化させる．ボタン群はクリックされることで JavaScript の関数が呼び出され，ボタンによってタグ挿入，プレビュー表示などの補助機能が実行される．

各々のテキストエリアでは，イベントリスナが入力されたキーの識別やテキストエリア内の文字列の変更検知を行なっている．改行など特定の文字が入力されると，メモリに現在のテキストエリアの文字列を格納する．記述者が ctrl+z を入力するなど，特定のキー入力をした場合にはメモリに保持している文字列でテキストエリアの文字列を上書きすることで，Undo や Redo の処理を行う．

記述者は記述用ページ内，もしくは別のタブにプレビュー画面を表示することができる．前者はテキストエリアとプレビューエリアのサイズを変更することで実現し，後者は記述用ページを親とする子ページを開くことで実現する．どちらも現在記述されているソースコードと解説を読み込み，閲覧用ページと同様の処理をその場で行うことでプレビューを生成する．

5. まとめ

近年のインターネットは大きな役割を担っており，プログラミングも同様だが，ソースコードと解説が主流に掲載されているサイトではソースコードと解説は上下に書かれているため，同時に読むことができない．CodeWiki[1] は上記の問題を解決したが，本文以外の情報も参照されてしまうために既存のページへの埋め込みに適していないため，ソースコードと解説をページの他の要素と同時に読むことはできず，新たなウィンドウやタブを開かなければならない．そこで本研究では，ソースコードと解説を同時に読むことができ，かつ既存のサイトに埋め込めるような Web サービスを提案する．

本研究の Web サービスでは埋め込みに対応した閲覧用ページと，記述を補助する記述用ページを提供する．閲覧用ページは既存のサイトに埋め込むことが可能なため，既に掲載されているソースコードや解説を，記事の他の情報と同時に読むことができるようになる．今後は実際の Web サービスへの埋め込みを試し，問題が出ないかなどを調査していく予定である．

参考文献

[1] Hiroaki Fukuda, “Toward efficient source code sharing on the Web, ” Proceedings of the 11th International Symposium on Open Collaboration, OpenSym ’15, ACM, 2015, pages 1-2.

[2] Anja Haake, Stephan Lukosch, and Till Schümmer. “Wiki-Templates: Adding Structure Support to Wikis On Demand, ” Proceedings of the 2005 International Symposium on Wikis , WikiSym ’05, ACM, 2005, pages 41-51.

[3] J. Gruber, “Darling fireball: Markdown”, https://daringfireball.net/projects/markdown ,2020-11-14 閲覧

[4]https://github.com/riversun/JSFrame.js